

POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering



**Politecnico
di Torino**

Master's Degree Thesis

A LoRaWAN[®] Tracking System for IoT

Supervisors

Prof. Claudio CASETTI

Ing. Mattia TRASFORINI

Candidate

Antonio EVA

JULY 2022

Summary

This thesis project was realised at Alten Italia R&D department in Bologna, with the intent of implementing an IoT asset-tracking system addressing the underlying principles of Industry 4.0. In global manufacturing and commerce, raw materials, components, and final goods can travel long distances, through warehouses and logistic hubs before arriving at their destination. The deployment of a fleet of programmable IoT devices able to track and store data of many unit loads provides a company valuable information for planning and optimization. This project studies the implementation steps around an IoT device capable to determine and send real-time position and condition of Unit Loads along the supply chain. LoRaWAN® has been chosen as communication network since it is considered one of the best low-power wide-area (LPWA) networks to wirelessly connect devices to the internet and perfectly meets IoT requirements such as bi-directional communication, end-to-end security, and localization services. The workbench to develop the architecture was The Things Network (TTN) that provides tools to manage end devices and gateways. TTN is a global collaborative IoT ecosystem used by thousands of developers and companies around the world and guarantees free access to LoRaWAN® services in more than 150 countries. The tracking device used is the prototype STEVAL-STRKT01 board produced by STMicroelectronics that fully responds to compactness, low cost and low power consuming requirements. It has been programmed to collect positioning, temperature, humidity, pressure, orientation data of a Unit Load through GNSS module and environmental sensors. The access to device data has been guaranteed by MQTT protocol and have been processed with Nodered, a powerful editor to create JavaScript functions. Data are stored on a database from which is possible to retrieve and analyse them on a dashboard. The system was tested through several outdoor tests to prove the scalability of the architecture and it can provide important information on the quality of transport service, reporting the coordinates of critical events, where sensor thresholds are exceeded.

Table of Contents

List of Tables	VI
List of Figures	VII
1 Introduction	1
1.1 Internet of Things	1
1.2 Industry 4.0	2
1.3 Logistics	4
1.4 Asset Tracking	5
1.5 Packaging	6
1.6 Purpose	8
1.7 Outline	8
2 LPWAN	9
2.1 What is LoRa?	9
2.2 LoRaWAN [®] Protocol	11
2.3 LoRaWAN [®] Architecture	12
2.3.1 End devices	12
2.3.2 Gateways	12
2.3.3 Network Server	13
2.3.4 Application Server	13
2.3.5 Join Server	14
2.4 Message Type	14
3 LoRa[®] Device	18
3.1 The STEVAL-STRKT01 End Node	18
3.2 Hardware description	18
3.3 Setup to Update the Firmware	20
4 Software implementation	23
4.1 STM32CubeIDE	23

4.2	Project structure	24
4.2.1	Drivers	24
4.2.2	Middlewares	24
4.3	Implemented Code	24
4.3.1	State Machine	25
4.3.2	Interrupts	26
4.3.3	GNSS satellite fix	26
4.4	Serial Port Communication	27
4.4.1	Over The Air Activation	27
5	The Things Network	29
5.1	Application	31
5.1.1	Device Registration	31
5.1.2	Gateway registration	31
5.2	Live Stream Message	32
5.2.1	Cayenne Payload	34
5.2.2	Payload Formatter	34
5.3	Transmission Limitation	36
5.3.1	Duty Cycle	36
5.3.2	TTN Fair Use Policy	36
5.4	MQTT Server	36
6	IoT platform	37
6.1	Node-RED	37
6.1.1	MQTT Client	37
6.1.2	Message parsing	38
6.1.3	Data Store	38
6.1.4	Load Status Check	38
6.1.5	Geofencing	41
6.1.6	Dashboard	41
6.1.7	Reporting and Notification	42
6.2	Database	43
7	Test and Experiments	45
7.1	TEST A: Mid-haul shipping through a non-urban environment . . .	45
7.2	TEST B: Urban delivery on Cargo Bike	50
8	Conclusion and Future Work	52
8.1	Conclusion	52
8.2	Future Work	52
	Bibliography	53

List of Tables

1.1	The importance of packaging optimization for cost saving	7
2.1	LoRa parameters	11
2.2	LoRa parameters	16
2.3	LoRa FPort values description	17
5.1	The Things Network in Italy	30
5.2	TTIG Registration Data	33
5.3	TTIG Datasheet	33
5.4	Cayenne Payload Format	34
5.5	Airtime Calculated	36
6.1	Variables from Uplink message	40
7.1	Load Alarm Sensor Thresholds	45
7.2	Test Description	45
7.3	Message data	47
7.4	Gateways	48
7.5	Threshold position	49
7.6	Load Alarm Thresholds	50
7.7	Test Description	51
7.8	Message data	51
7.9	Threshold Position	51

List of Figures

1.1	IoT	1
1.2	Worldwide spending on the IoT from 2018 to 2023[1]	2
1.3	Steps of industrial development Industry 4.0.	3
1.4	Market size of Industry 4.0 in Italy [2].	3
1.5	RFID technology	4
1.6	Global IoT in Logistics Market, 2016-2026 [5]	5
1.7	Asset tracking	5
1.8	An example of Drop Tower	7
2.1	Bandwidth vs Range	9
2.2	LoRaWAN [®] technology stack	11
2.3	LoRaWAN network architecture	12
2.4	Gateway overview	13
2.5	Uplink/Downlink transmission	14
2.6	Device classes and Energy Consumption	16
2.7	LoRaWAN MAC frame. The sizes of the fields are in bits	17
3.1	STEVAL-STRKT01 evaluation board	18
3.2	STEVAL-STRKT01 components, topview	19
3.3	Firmware update setup	21
3.4	STM32 Nucleo Board	22
4.1	STM32CubeIDE logo	23
4.2	Function pack FP-ATR-LORA1	24
4.3	Flowchart	25
4.4	Tera Term Serial Port Configuration	27
4.5	Serial USB terminal Android App	28
4.6	OTAA LoRa keys	28
5.1	The Things Network in Italy	29
5.2	The Things Network Application	31
5.3	An example of outdoor LoRa Gateway models	32

5.4	The Things Indoor Gateway	32
5.5	Live Data TTN Application Console	33
5.6	Cayenne Payload Format	34
5.7	Payload Hex format	35
5.8	Payload JSON format	35
6.1	Raspberry Pi 4 - 1GB RAM	37
6.2	MQTT-in node	38
6.3	Uplink Message	39
6.4	MQTT-in node configuration	39
6.5	Device axis	40
6.6	Geofence	41
6.7	Dashboard Node-RED flow	42
6.8	Dashboard Indicators	42
6.9	Dashboard graphs	43
6.10	Device Data retrieved from the Database	44
6.11	Connection Data retrieved from the Database	44
7.1	Device position (yellow) and Gateways position (red)	46
7.2	Distance Device - Gateway	49
7.3	cargo bike set-up	50
7.4	Device position (yellow) and Gateways position (red)	51

Chapter 1

Introduction

1.1 Internet of Things

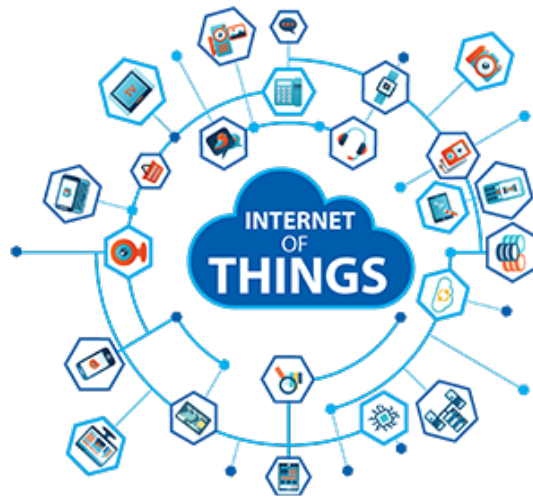


Figure 1.1: IoT

The Internet of Things can be referred as a multitude of physical objects wirelessly connected to a network and spread over areas of difficult access to acquire different type of information. These devices are generally composed by sensors, powered by batteries, and they generate and periodically send small amount of useful data to the network. The estimated number of connected devices worldwide was 8 billion in 2020 and it expected to triple by 2030, reaching more than 25 billion [1].

There are three fundamental enablers for an IoT architecture:

1. Universal communication networks.
2. Powerful and easy-to-integrate sensors and actuators.
3. Cloud services for processing large amounts of data.

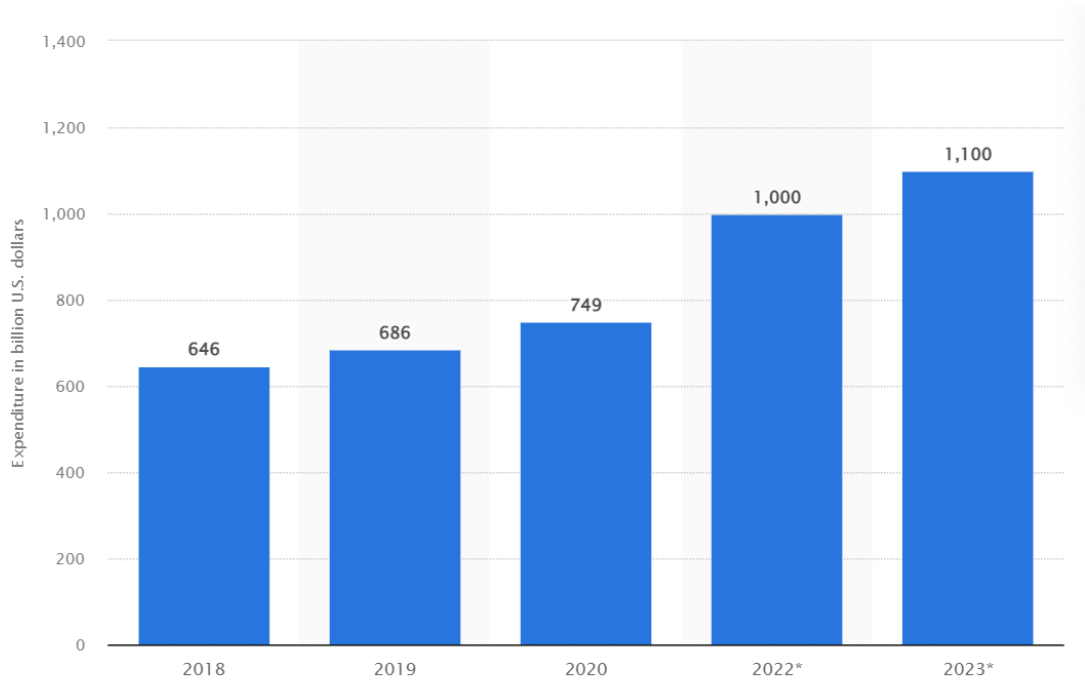


Figure 1.2: Worldwide spending on the IoT from 2018 to 2023[1]

1.2 Industry 4.0

The introduction of the Internet of Things into the industrial field is a key factor for the upcoming fourth industrial revolution: Industry 4.0. Factories are developing environments in which the gap between real and digital world is becoming smaller, by connecting all the assets involved in the supply chain. Growing number of companies are implementing Industrial IoT (IIoT) solutions to convert their real-time and historical data into meaningful business knowledge to improve efficiency and increase competitiveness.

The American Society for Quality (ASQ) conducted a survey showing that manufacturing companies that have digitized their operations with IoT solutions increase in production efficiency, with a 10% reduction in downtime and 49% fewer product defects [3]. The IIoT can transform traditional supply chains into

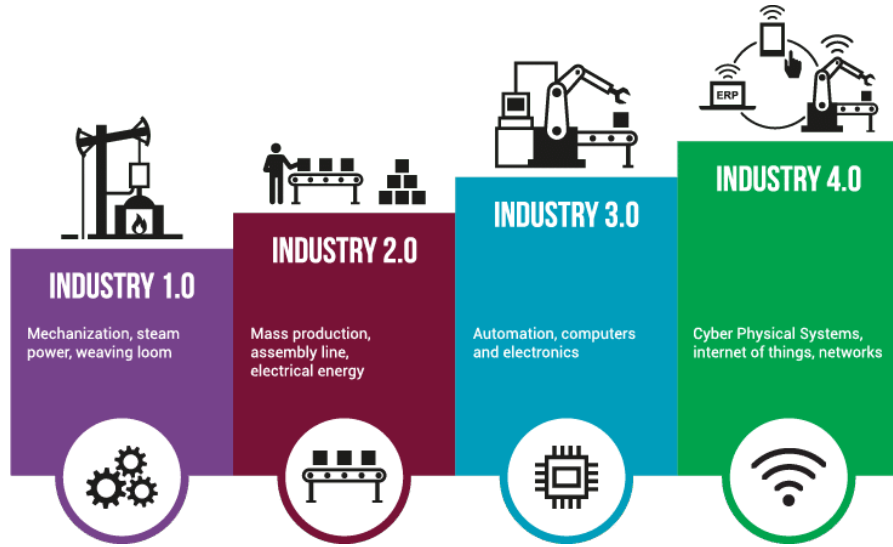


Figure 1.3: Steps of industrial development Industry 4.0.

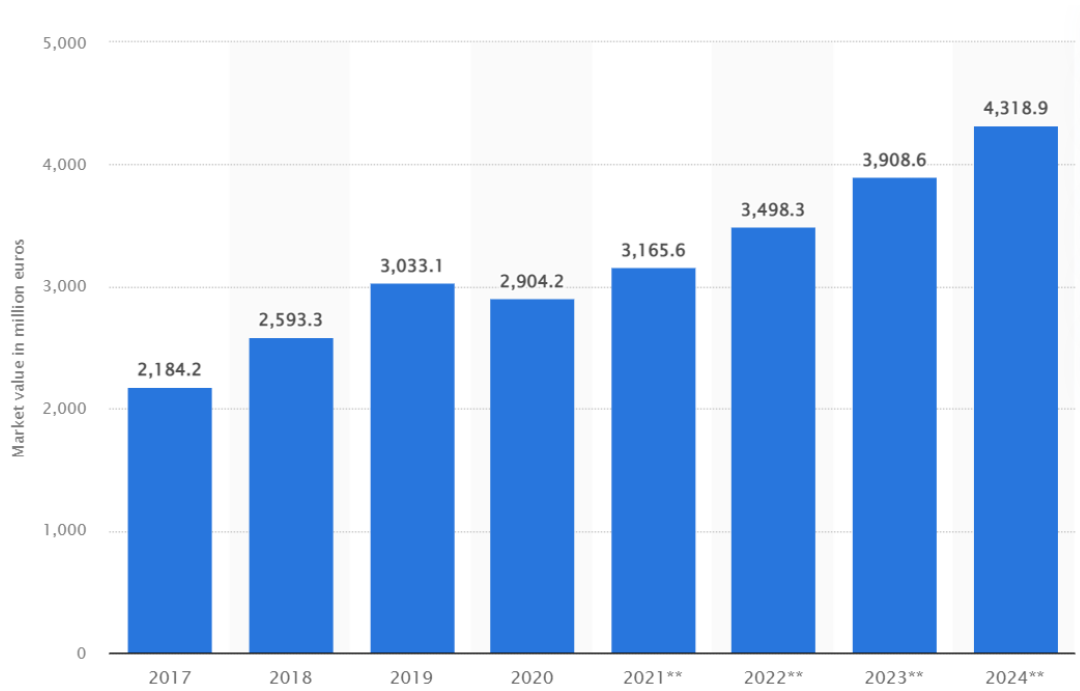


Figure 1.4: Market size of Industry 4.0 in Italy [2].

dynamic and interconnected systems, according to the definition of Smart Factory. An important aspect of the Industry 4.0. is the implementation of predictive maintenance solutions to avoid downtime in the operational processes. IoT allows

to collect and analyse data from multiple sensors and to be proactive about risk-management, increasing resilience of the supply chain. With the launch of the National Industry 4.0 Plan in 2017, the investments on the Italian market was expected to reach 3.1 billion euros[2].

1.3 Logistics

Logistics refers to the overall process of planning, managing, and controlling the flow of goods from the supplier to the customer.

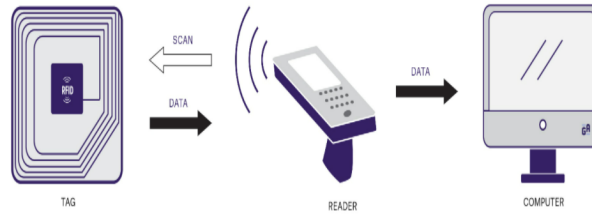


Figure 1.5: RFID technology

One of the main goal of distribution is to reduce the costs maintaining the level of service provided and the information flow plays a fundamental role in this sense, connecting various stages of the supply chain to coordinate their actions and optimize the performance. The principal technologies to identify the goods and monitor the assets is the RFID (Radio-frequency Identification), fundamental for warehouses management. In the early 90's, wireless devices began to be used and then in the 2000's, the advent of the internet was the turning point for information management. From 2009 the IoT started gaining moment and drastically altered the industrial and logistics applications. IoT size in the logistics worldwide market was \$34,504.8 million in 2019, and it is expected to grow with a CAGR (Compound annual growth rate) of 13.2% between 2020–2030 [4]. The explosion of e-commerce caused by the 2020 health emergency, made clear the need to strengthen distribution and made more efficient logistics operations. The implementation of IoT in logistics gives a rapid and effective result combining extensive sensing, satellite-based positioning, wireless communication, and cloud platforms for data storage, processing, and visualization. Monitoring the logistics with IoT solutions provides a large amount of data to define key performance indicators (KPIs).

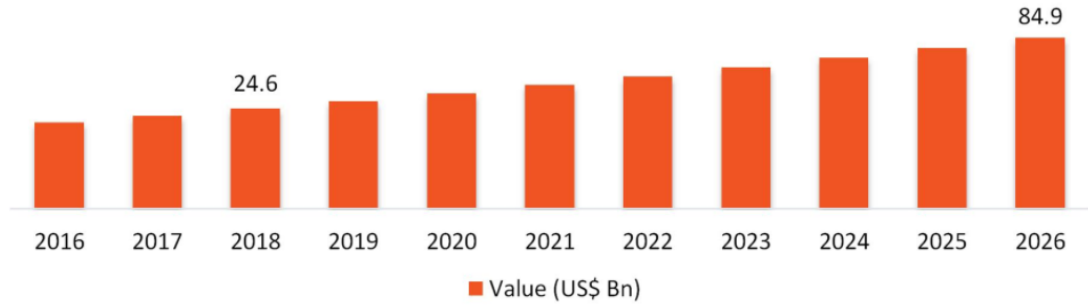


Figure 1.6: Global IoT in Logistics Market, 2016-2026 [5]

1.4 Asset Tracking

Tracking its physical assets, a company increase the transparency of the delivery process of products from the warehouse to the customer to verify that all stages of the supply chain are completed smoothly. Data as temperatures, shocks and tilts

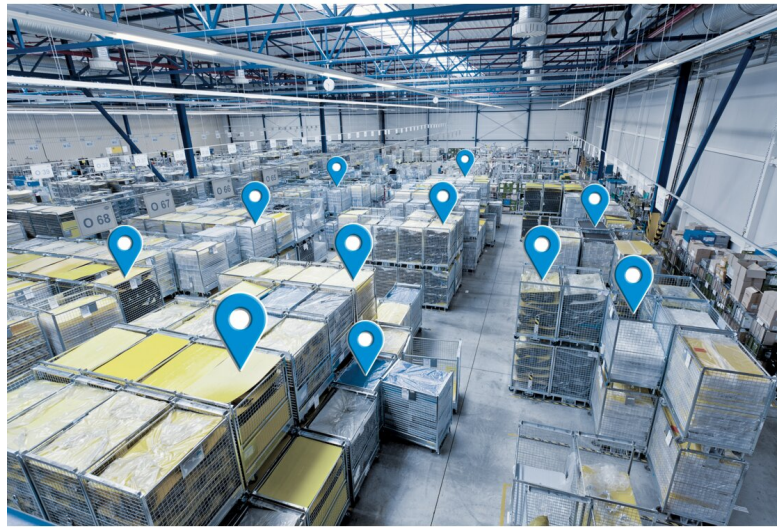


Figure 1.7: Asset tracking

of goods provides better insights into quality control, monitoring if materials are delivered safely, on time and in perfect condition. Tracking high-value, fragile, and sensitive packages can be very convenient for three reasons:

- Customer service: access to real-time data keeps the receiving customer

informed.

- Dispute resolution: hard data will lead to reduced insurance premiums.
- Network optimization: costly bottlenecks or locations at which specific incidents are occurring with shipments can be easily identified.

1.5 Packaging

The design of products must take account of the logistics necessary for its distribution. In many production sectors there is a shift from "Design for product" to "Design for logistics" [6]. All products packed during their distribution are subjected to different stresses as shocks, vibrations, high or low temperature and humidity. There are three phases of the distribution where IoT can be a solution for stresses analysed:

- Handling phase: damage caused by falls.
- Storage phase: static and dynamic compression loads
- Transportation phase: dynamic stresses strongly related with the handling vehicle used.

The Fragility Index is an important parameter for packaging design and takes into account the whole life of an object, including transportation. It is an dimensionless quantity defined as:

$$G = a/g \quad (1.1)$$

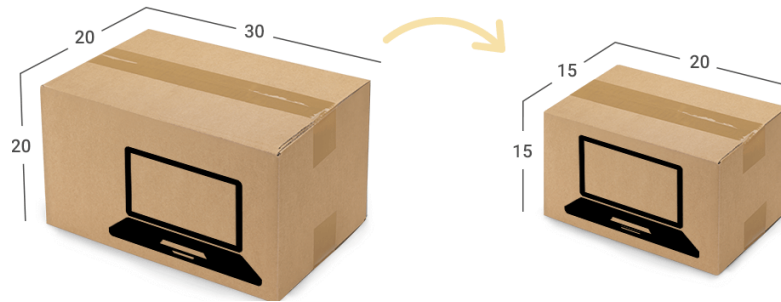
where a = max acceleration bearable and g = gravitational acceleration. The fragility of a product can be determined through several laboratory tests, such as drop towers used to measure energy absorbed of a sample or shock machines.

A correct packaging process must consider any degradation during the transition from manufacturing to the stage of consumption. Thanks to IoT, packaging is becoming a quality control platform focused on a variety of consumer benefits such as product quality, product safety and freshness. IoT collect very useful data along the supply to strongly support packaging optimization bringing different benefits for the supply chain:

- Cost Savings: Avoiding oversize packaging can lead to tremendous savings throughout the supply chain on material costs and on weight for transportation.
- Reduction in Damages: Choosing the optimal packing materials leads to a reduction in damages.



Figure 1.8: An example of Drop Tower



	Packaging (A)	Packaging (B)
Product weight	1kg	1kg
Dimensional weigh	$(20 \times 20 \times 30) / 5000 = 2,4$	$(15 \times 15 \times 20) / 5000$
Billable weight	2,4 kg	1 kg

Table 1.1: The importance of packaging optimization for cost saving

- **Minimal Carbon Footprint:** Materials like plastic, Styrofoam, and even cardboard are proven to be harmful to the environment as they produce a lot of waste and take years to break down. Reducing the size of the packaging to the minimum, decrease the carbon footprint, help the environment and makes consumers more satisfied.
- **Effective Delivery:** Thousands of dollars can be saved every year rethinking product designs from a delivery perspective.

1.6 Purpose

The topic of this thesis is to implement a IoT tracking system able to collect position and environmental data of Unit Loads with the aim of reporting the stresses suffered by them along the supply chain. An evaluation board (STEVAL-STRKT01), produced by STMicroelectronics, has been programmed to read temperature and humidity sensors and acquire positioning data. Placing the device inside a Unit Load, the goal was to acquire real time data and send alerts when thresholds on the accelerometer and on the environment sensors were exceeded. The LoRaWAN[®] communication protocol has been used to periodically transmit the data collected by the device and a dashboard has been configured to display them. Particular attention was given to limit Transmission Length, Payload size, and Duty Cycle in order to compliance with LoRaWAN[®] limitations. The device proved to be reliable during the tests conducted and to be a valid solution for large-scale applications.

1.7 Outline

This thesis is composed by eight chapters: Introduction, LPWAN, LoRa Device, Software implementation, The Things Networ, IoT Platform, Test and Experiments, CONclusion.

Chapter 2

LPWAN

Low-power wide area network (LPWAN) is a technology that interconnects low-bandwidth, battery-powered devices with low bit rates over long ranges. Created for machine-to-machine (M2M) and internet of things (IoT) networks, LPWANs operate at a lower cost with greater power efficiency than traditional mobile networks. They are also able to support a greater number of connected devices over a larger area.

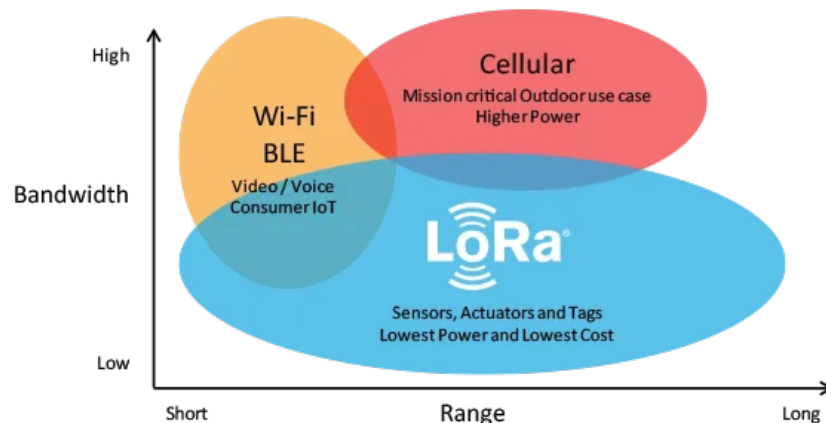


Figure 2.1: Bandwidth vs Range

2.1 What is LoRa?

LoRa is one of the most important wireless modulation technique currently used around the world and for its characteristics is the perfect technology for the development and use of IoT devices. LoRa is derived from Chirp Spread Spectrum

(CSS), developed for radar applications in the 1940's that consist of a linear variation in frequency, called chirp, which ranges over the entire band assigned to the channel (typically 125, 250 or 500 KHz). In Europe the bandwidth for LoRa application is limited to 125 and 250kHz. In LoRa modulation the spreading of the spectrum is achieved by generating a chirp signal that continuously varies in frequency. An advantage of this method is that timing and frequency offsets between transmitter and receiver are equivalent, greatly reducing the complexity of the receiver design. The frequency bandwidth of this chirp is equivalent to the spectral bandwidth of the signal. The wanted data signal is chipped at a higher data rate and modulated onto the chirp signal. LoRa can be operated on the license free sub-gigahertz bands, for example, 915 MHz, 868 MHz, and 433 MHz [7].

The relationship between the wanted data bit rate, symbol rate and chip rate for LoRa modulation can be expressed as follows. We can define the modulation bit rate R_b as:

$$R_b = SF * \frac{1}{2^{SF}} \frac{bits}{secs} \quad (2.1)$$

Where SF = spreading factor (from 7 to 12), i.e. the bit size of the symbol and BW = modulation bandwidth (Hz). The period is defined as:

$$T_s = \frac{2^{SF}}{BW} secs \quad (2.2)$$

The symbol rate is:

$$R_s = \frac{1}{T_s} = \frac{BW}{2^{SF}} symbol/sec \quad (2.3)$$

In the end the chip rate is:

$$R_c = R_s * 2^{SF} = BW chips/sec \quad (2.4)$$

LoRa modulation also includes a variable error correction scheme that improves the robustness of the transmitted signal at the expense of redundancy.

Thus the nominal bit rate for the data signal is:

$$R_b = SF * \frac{\frac{4}{4+CR}}{2^{SF}} \frac{bits}{sec} \quad (2.5)$$

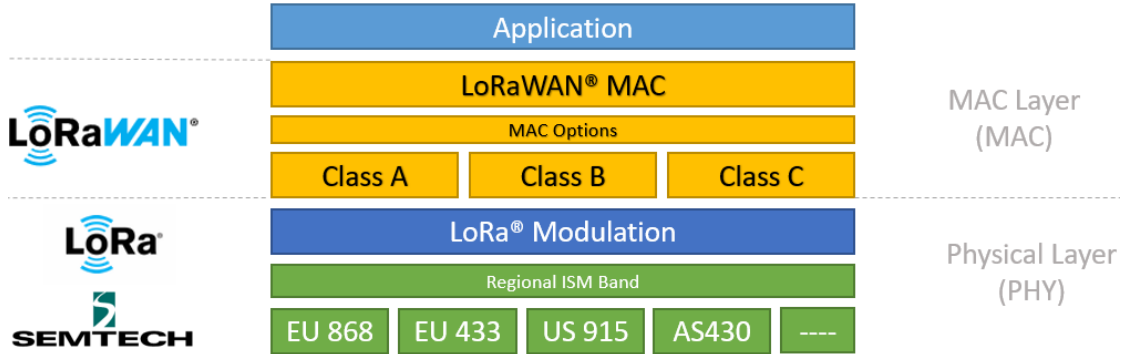
Where CR = code rate (from 1 to 4).

Spreading Factor	Coding Rate	Bandwidth (kHz)	Nominal Bitrate (bps)
8	4/5	125	5470
8	4/5	250	11000
12	4/5	125	293
12	4/5	250	586

Table 2.1: LoRa parameters

2.2 LoRaWAN[®] Protocol

LoRaWAN[®] is a Media Access Control (MAC) layer protocol built on top of LoRa modulation. It is a software layer which defines how devices use the LoRa hardware, for example when they transmit, and the format of messages. The LoRaWAN protocol is developed and maintained by the LoRa Alliance[®].

Figure 2.2: LoRaWAN[®] technology stack

Designed to wirelessly connect battery operated things to the internet in regional, national or global networks, the LoRaWAN[®] standard meets key Internet of Things (IoT) requirements such as bidirectional communication, end-to-end security, mobility, and geolocation services. The specification defines the device-to-infrastructure of LoRa[®] physical layer parameters and the LoRaWAN[®] standard, and provides seamless interoperability between manufacturers. While Semtech provides LoRa chipsets, the LoRa Alliance[®] drives the standardization and global harmonization of the LoRaWAN[®] standard for the vast ecosystem [8].

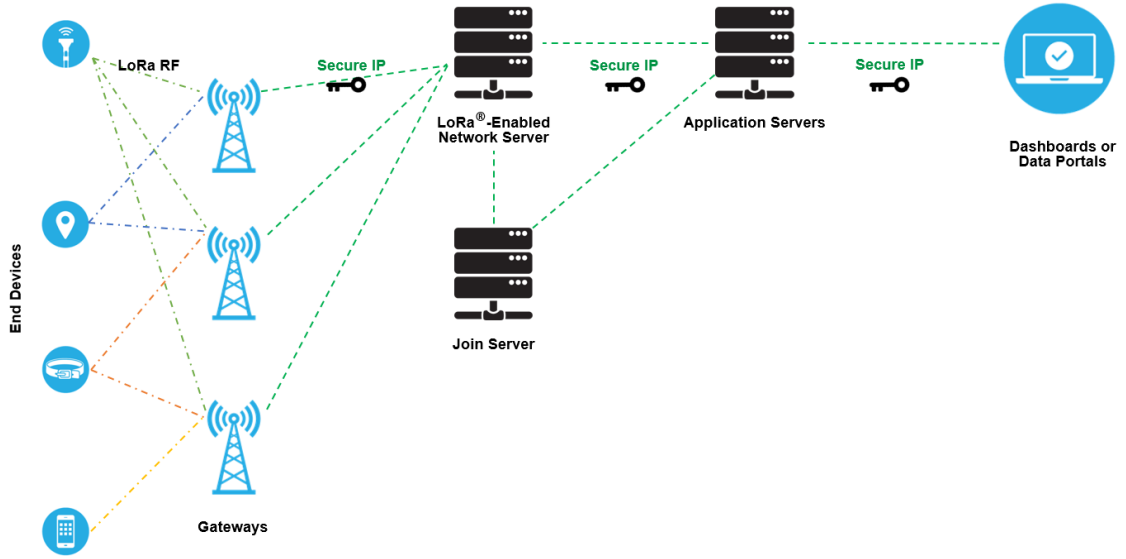


Figure 2.3: LoRaWAN network architecture

2.3 LoRaWAN[®] Architecture

The LoRaWAN[®] protocol defines four main components within the network: end device, gateway and network server.

2.3.1 End devices

Devices placed at the end of the network, equipped with sensors to periodically collect and monitor the environment. They usually come in the form of a low-power microcontroller that can be used in the field for many years without any maintenance and equipped with a low-power LoRa transmitter to send data packets. They are designed for IoT communication.

2.3.2 Gateways

LoRaWAN[®] gateways are the bridge between nodes and the network. They receive information from the end nodes using a LoRa concentrator and then forward the data to the network server via the Internet or a private network infrastructure. A gateway is able to listen to broadcasts on even more channels and to decode packets sent with a different spreading factor at the same time. LoRaWAN[®] gateways can be categorized into indoor and outdoor gateways. Indoor gateways are cost-effective and suitable for providing coverage in places as basements and multi-floor buildings. Outdoor gateways provide a larger coverage than the indoor gateways. They are

suitable for providing coverage in both rural and urban areas. It depends on the gateway specifications which is usually defined by the number of packets it can support. A single eight-channel gateway can support a few hundred thousand messages over the course of a 24-hour period. If each end device sends 10 messages a day, such a gateway can support about 10,000 devices [9].

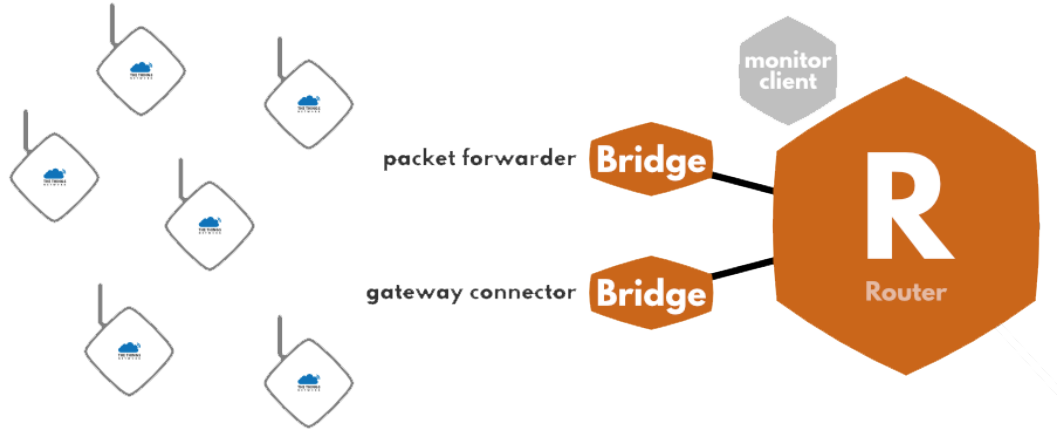


Figure 2.4: Gateway overview

2.3.3 Network Server

The network server is a back-end component that carries out multiple processes complex in relation to network management. It is responsible for the reception of data from the various gateways and performs various functions including eliminating any duplicate packets. The network server implements the adaptive data rate (ADR) functionality for the purpose of maximize the life of the batteries that power the devices and the capacity total network. The network server assigns each end node you want connect to the grid, a data rate and an output power to use for RF transmission, different for every need. The ADR algorithm assigns a higher data rate to the terminal nodes closest to the gateway since less susceptible to interference, and a lower output power for the RF transmission. The Things Network provides a LoRaWAN Network Server used by thousands of companies and developers around the world, it securely manages applications, end devices and gateways and is built by The Things Industries.

2.3.4 Application Server

The Application Server handles the LoRaWAN[®] application layer, including uplink data decryption and decoding, downlink queuing and downlink data encoding and

encryption.

2.3.5 Join Server

The Join Server handles the LoRaWAN[®] join flow, including Network and Application Server authentication and session key generation. In case a join-request is accepted, the Join Server derives session security context, which contains the session keys and is identified by a session key ID. Join Servers encrypt derived network and application session keys using key encryption shared between Network Servers and Application Servers respectively and include the session keys in the join-accepts in encrypted form [10].

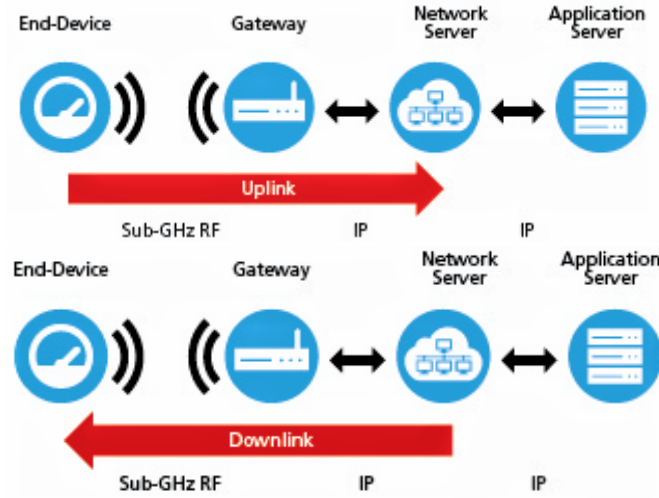


Figure 2.5: Uplink/Downlink transmission

2.4 Message Type

At the moment the LoRaWAN[®] specifications have reached version 1.1, but it should be noted that a LoRaWAN[®] 1.1 network is backward compatible and able to interoperate even with legacy LoRaWAN[®] 1.0.x devices. The LoRa Alliance also provides LoRaWAN[®] certification for devices which ensures interoperability and compliance on any LoRaWAN network, confirming that the device meets the functional requirements of the LoRaWAN[®] protocol specifications. The LoRaWAN[®] network architecture uses a star topology in which each end node communicates with multiple gateways that communicate with the network server. The MAC layer originally implemented on each LoRa device is based on the pure ALOHA protocol, that is, as previously mentioned, the nodes send messages in Uplink to

all the listening gateways, establishing a broadcast communication. There are two type of transmission between an end-device and a gateway:

- Uplink: from end-device to gateway.
- Downlink: from gateway to end-device.

While end devices can always send uplinks at will, the device's class determines when it can receive downlinks. By default LoRa devices operate in class A mode with energy saving properties, while they must be explicitly configured to operate in class B or C. The class also determines a device's energy efficiency.

- Class A, bidirectional: Class A devices can schedule sending an uplink transmission according to application specifications at regular time intervals. These types of devices allow two-way communications through a mechanism for which remain listening for a certain time interval of any downlink broadcasts only after having performed a broadcast in uplink. The time they are listening is determined by two short receiving windows after transmitting a packet. Once finished the reception time the class A device returns to mode dormant to conserve battery power.
- Class B, bidirectional with scheduled receive slots: These types of devices listen for downlink communications by opening a window in reception at scheduled time intervals. Because of this type of communication and a synchronization mechanism is required with the gateway that takes place via beacons. The network server will have to know when the end device will be listening.
- Class C, bidirectional with with reception slot: To this category belong all those devices that are constantly listening. The receiver LoRA remains active allowing immediate reception without that the device opens a listening window.

The choice of the end device class must be weighted according to the requirements application and the context in which it operates. Classes A and B guarantee improved energy efficiency and are designed for devices battery powered, while class C end devices are powered through electrical grid. This type of communication is not meant for the energy saving but ensures that the device can receive directives at any time. LoRaWAN defines several MAC message types. The following table presents MAC message types that can be found in LoRaWAN 1.0.x and 1.1 to send both MAC commands and application data. A data message can be composed by both MAC commands and application data. MAC payload of the data messages consists of a frame header (FHDR) followed by an optional port field (FPort) and an optional frame payload (FRMPayload). MAC commands can be sent either in the frame options field (FOpts) field or frame payload field (FRMPayload) field

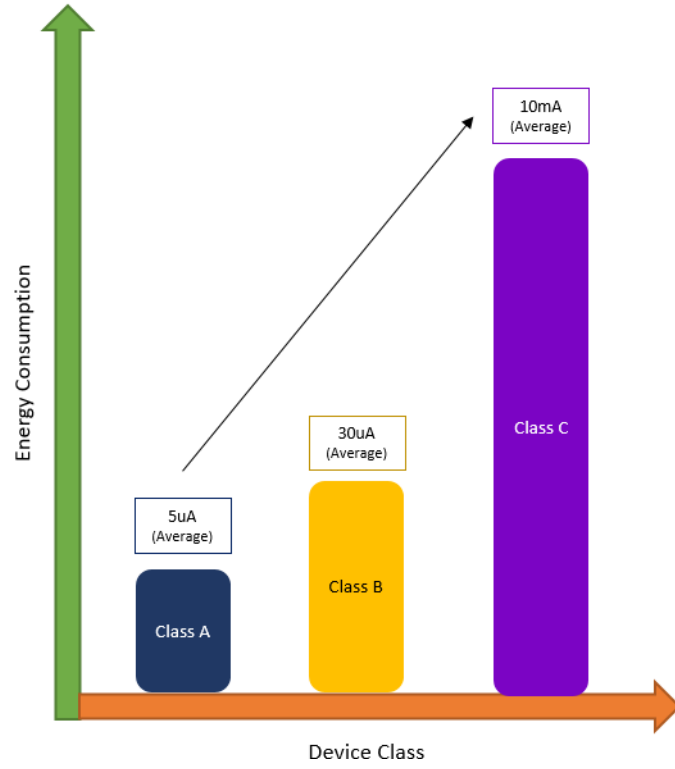


Figure 2.6: Device classes and Energy Consumption

LoRaWAN 1.0.x	LoRaWAN 1.1	Description
Join-request	Join-request	Uplink message
Join-accept	Join-accept	Downlink message
Unconfirmed Data Up	Unconfirmed Data Up	Uplink message
Confirmed Data Up	Confirmed Data Up	Uplink message
Confirmed Data Down	Confirmed Data Down	Downlink message
Reserved for future use	Rejoin-request Down	Uplink message

Table 2.2: LoRa parameters

of a data message, but not both simultaneously. Application data can be sent in the frame payload (FRMPayload) field of a data message. The FRMPayload field CAN NOT contain MAC commands and application data simultaneously.

The FPort field indicates if the message contains only MAC commands or application data. The Message Integrity Code (MIC) is the method to control the integrity of a message.

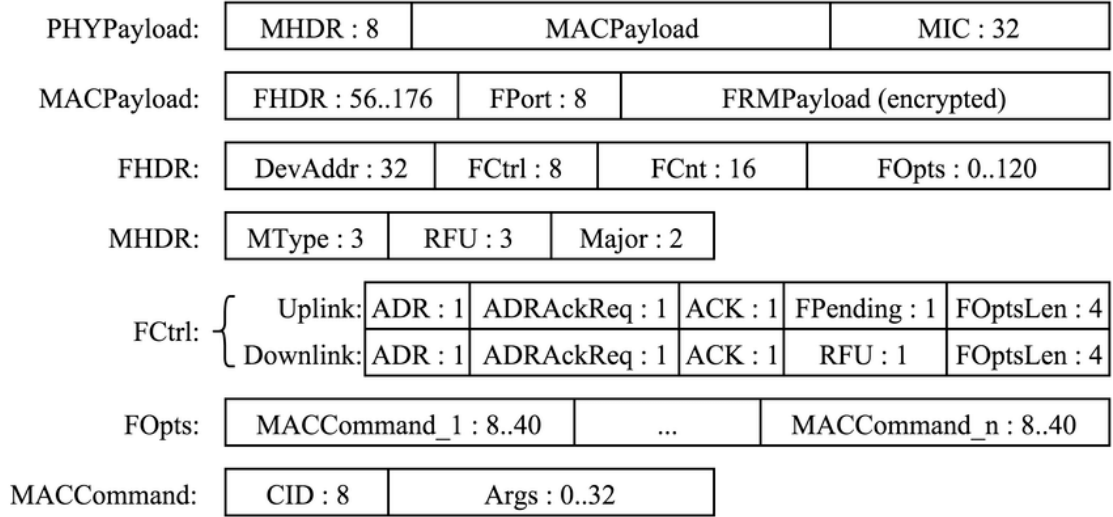


Figure 2.7: LoRaWAN MAC frame. The sizes of the fields are in bits

FPort value	Description
0	MAC command only
From 1 to 223	Application specific data
224	LoRaWAN MAC layer test protocol

Table 2.3: LoRa FPort values description

Chapter 3

LoRa[®] Device

3.1 The STEVAL-STRKT01 End Node



Figure 3.1: STEVAL-STRKT01 evaluation board

Alten Italia provides developers with devices produced by STMicroelectronics for research and training projects on Embedded systems. The device must comply with compactness and resistance requirements for the development of this thesis project, guaranteeing easy prototyping and development of a solution capable of interfacing with the LoRa network and with the GNSS system. The choice fell on the STEVAL-STRKT01 evaluation board, hardware recently on the market and with a low price (80€-100€) whose main features are listed below.

3.2 Hardware description

The STEVAL-STRKT01 is made up of the following components:

- STM32L072CZ- ultra-low-power ARM Cortex-M0+ MCU: The ultra-low-power STM32L072xx microcontrollers incorporate the connectivity power of the universal serial bus (USB 2.0 crystal-less) with the high-performance Arm

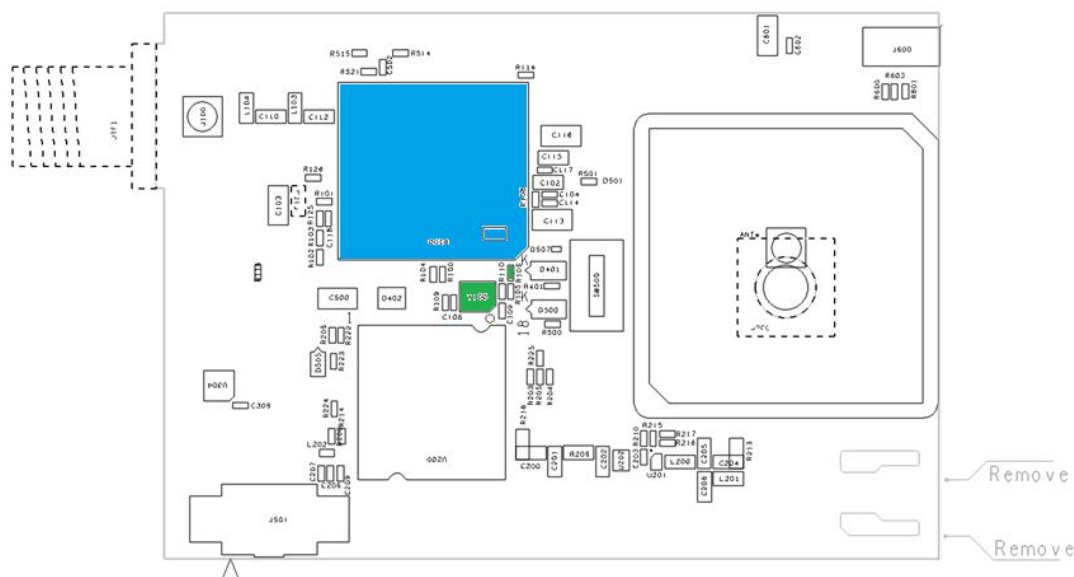


Figure 3.2: STEVAL-STRKT01 components, topview

Cortex-M0+ 32-bit RISC core operating at a 32 MHz frequency, a memory protection unit (MPU), high-speed embedded memories (up to 192 Kbytes of Flash program memory, 6 Kbytes of data EEPROM and 20 Kbytes of RAM) plus an extensive range of enhanced I/Os and peripherals.

- Teseo-LIV3F GNSS module: The Teseo-LIV3F module is an easy to use Global Navigation Satellite System (GNSS) standalone module, embedding TeseoIII single die standalone positioning receiver IC working simultaneously on multiple constellations (GPS/Galileo/Glonass/BeiDou/QZSS). 9.7x10.1 mm tiny size.
- STBC02 - Li-Ion linear battery charger: The STBC02 is a highly integrated power management, embedding a linear battery charger, a 150 mA LDO, 2 SPDT load switches, a smart reset/watchdog block and a protection circuit module (PCM) to prevent the battery from being damaged under fault conditions. The STBC02 uses a CC/CV algorithm to charge the battery; the fast charge and the pre-charge current can be both independently programmed using dedicated resistors. The termination current is set by default, being 5% of the programmed fast charge current, but it can also be fixed to different values. Likewise, the battery floating voltage value is programmable and can be set to a value up to 4.45 V. The STBC02 also features a charger enable input to stop the charging process anytime.
- M95M02-DR - 2 Mbit serial SPI bus EEPROM: The M95M02 devices are

Electrically Erasable PROgrammable Memories (EEPROMs) organized as 262144 x 8 bits, accessed through the SPI bus. Over an ambient temperature range of -40 °C / +85 °C the M95M02-DR can operate with a supply voltage from 1.8 to 5.5 V.

- CMWX1ZZABZ-091 LoRa module (by Murata): The STEVAL-STRKT01 embeds the CMWX1ZZABZ-091 LoRa[®]/Sigfox[™] module (by Murata) that allows easily developing applications with the STM32L072CZ and the LoRa[®]/Sigfox[™] RF connectivity in one single module.
- HTS221 is an ultra-compact sensor for relative humidity and temperature. It includes a sensing element and a mixed signal ASIC to provide the measurement information through digital serial interfaces. The sensing element consists of a polymer dielectric planar capacitor structure capable of detecting relative humidity variations and is manufactured using a dedicated ST process.
- LIS2DW12: is an ultra-low-power high-performance three-axis linear accelerometer belonging to the “femto” family which leverages on the robust and mature manufacturing processes already used for the production of micromachined accelerometers. It has user-selectable full scales of $\pm 2g/\pm 4g/\pm 8g/\pm 16g$ and is capable of measuring accelerations (with output data rates from 1.6 to 1600 Hz) and an integrated 32-level first-in, first-out (FIFO) buffer allowing the user to store data to limit intervention by the host processor.
- The LPS22HB is an ultra-compact piezo resistive absolute pressure sensor which functions as a digital output barometer. The device comprises a sensing element and an IC interface which communicates through I²C or SPI from the sensing element to the application. The sensing element, which detects absolute pressure, consists of a suspended membrane manufactured using a dedicated process developed by ST. The LPS22HB is available in a full-mold, holed LGA package (HLGA). It is guaranteed to operate over a temperature range extending from -40 to +85 °C [11].

3.3 Setup to Update the Firmware

The procedure chosen to flash the code on the STEVAL-STRKT01 involves the use of a STM32 Nucleo-64 because integrates the debugger and programmer ST-LINK and ensure digital isolation between the PC and the target evaluation board.

- Step 1: Connect 5 jumper wires (male side) to the CN501 connector of the STEVAL-STRKT01

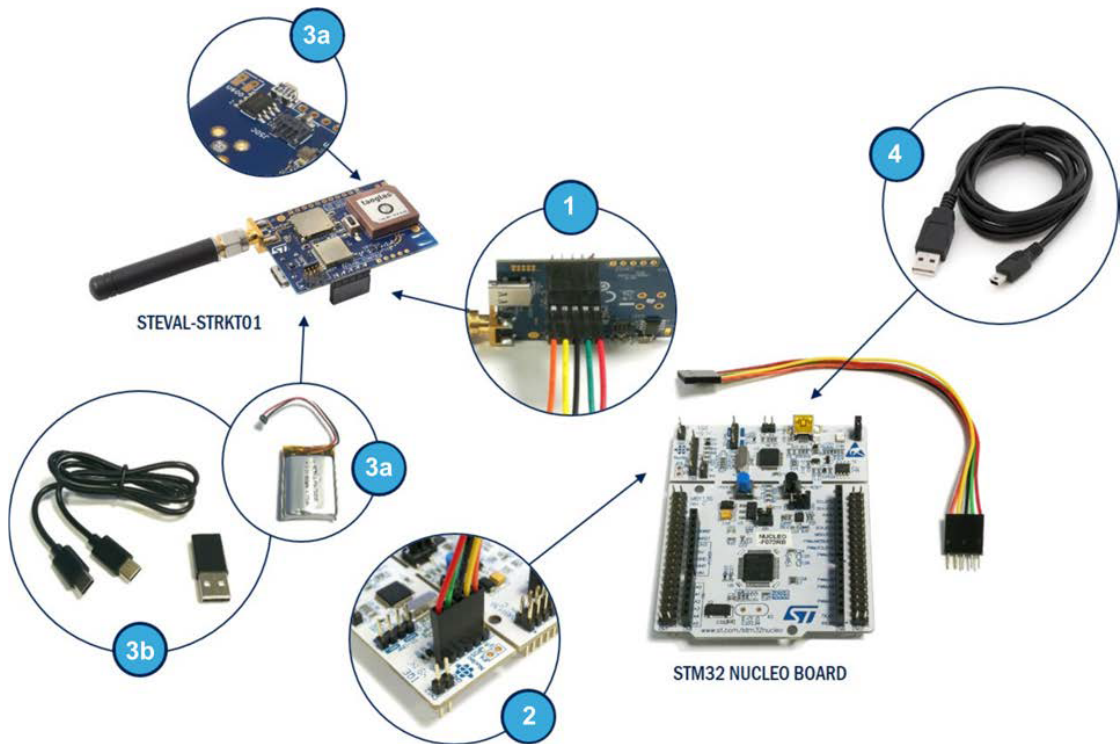


Figure 3.3: Firmware update setup

- Step 2: Connect the female side of the jumper wires to the SWD connector of the STM32 Nucleo Board, the 6th pin must be unconnected.
- Step 3a: Connect the battery
- Step 3b: Plug the Type-C USB cable in the PC and supply the STEVAL-STRKT01.
- Step 4: Connect the STM32 Nucleo Board to a PC through Type-A/mini B cable

The STM32 Nucleo-64 board provides an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features, provided by the STM32 microcontroller.

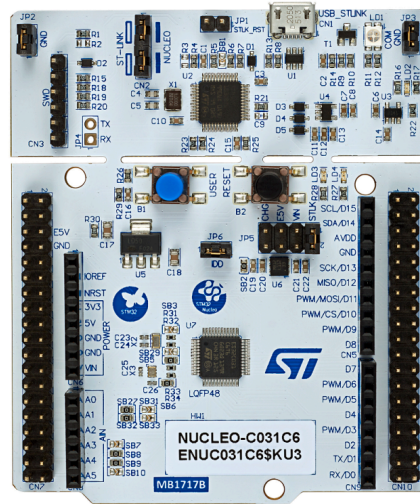


Figure 3.4: STM32 Nucleo Board

Chapter 4

Software implementation

4.1 STM32CubeIDE

STMicroelectronics increases the functionality of the energy-efficient STM32 microcontrollers offering the all-in-one solution STM32CubeIDE. It is offered under an open license and has dedicated STM32-specific features, which can simplify and accelerate embedded projects. The powerful STM32CubeMX tool to

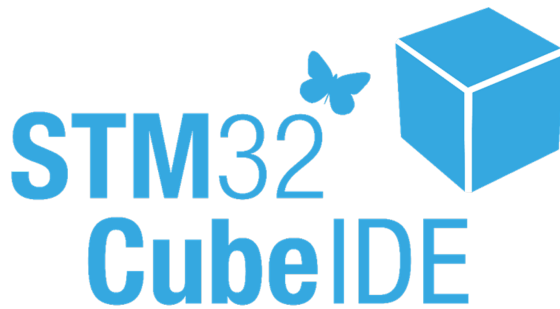


Figure 4.1: STM32CubeIDE logo

configure the microcontroller and manage the construction of the project. The STM32Cube ecosystem is very popular with developers, averaging over 250,000 downloads of the STM32CubeMX tool per year. By combining STM32CubeMX with STM32CubeIDE, a much more powerful development environment is created than the typical free tools offered by MCU resellers. The complete STM32Cube ecosystem also contains the STM32CubeProgrammer for programming MCUs and the STM32CubeMonitor series for monitoring application behavior, as well as individual software packages specific to MCUs.

4.2 Project structure

STMicroelectronics supplies together with the STEVAL-STRKT01 device a software package for STM32 microcontrollers that allows interfacing with environmental sensors, Teseo GNSS module and with the LoRa network [12].

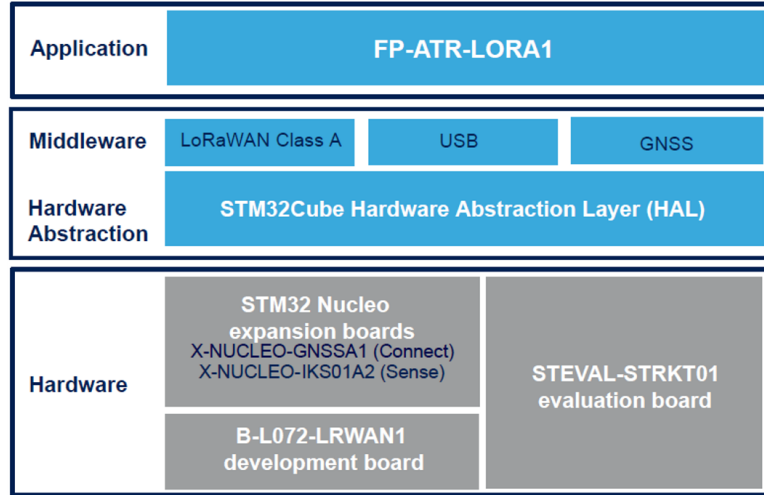


Figure 4.2: Function pack FP-ATR-LORA1

4.2.1 Drivers

The STM32 Hardware Abstraction Layer (HAL) provides a simple, generic multi-instance set of APIs (application programming interfaces) to interact with the upper layers like the user application, libraries and stacks.

4.2.2 Middlewares

Libraries and protocols to communicate with sensors.

4.3 Implemented Code

In this section an overview of the implemented code will be made. STMicroelectronics FP-ATR-LORA1 function pack comes with a basic software used as starting point to develop this project. The main.c file has been modified to define the device behaviour.

4.3.1 State Machine

The device can work in different states to optimize battery discharge and be compliant to LoRaWAN Fair Access Policy. When a unit load remains stored for a long time in a warehouse the device can read sensors less frequently and lengthen the time between sending messages, reducing the Duty Cycle. The device wakes up and track the unit load more accurately when a sensor interrupt occurs.

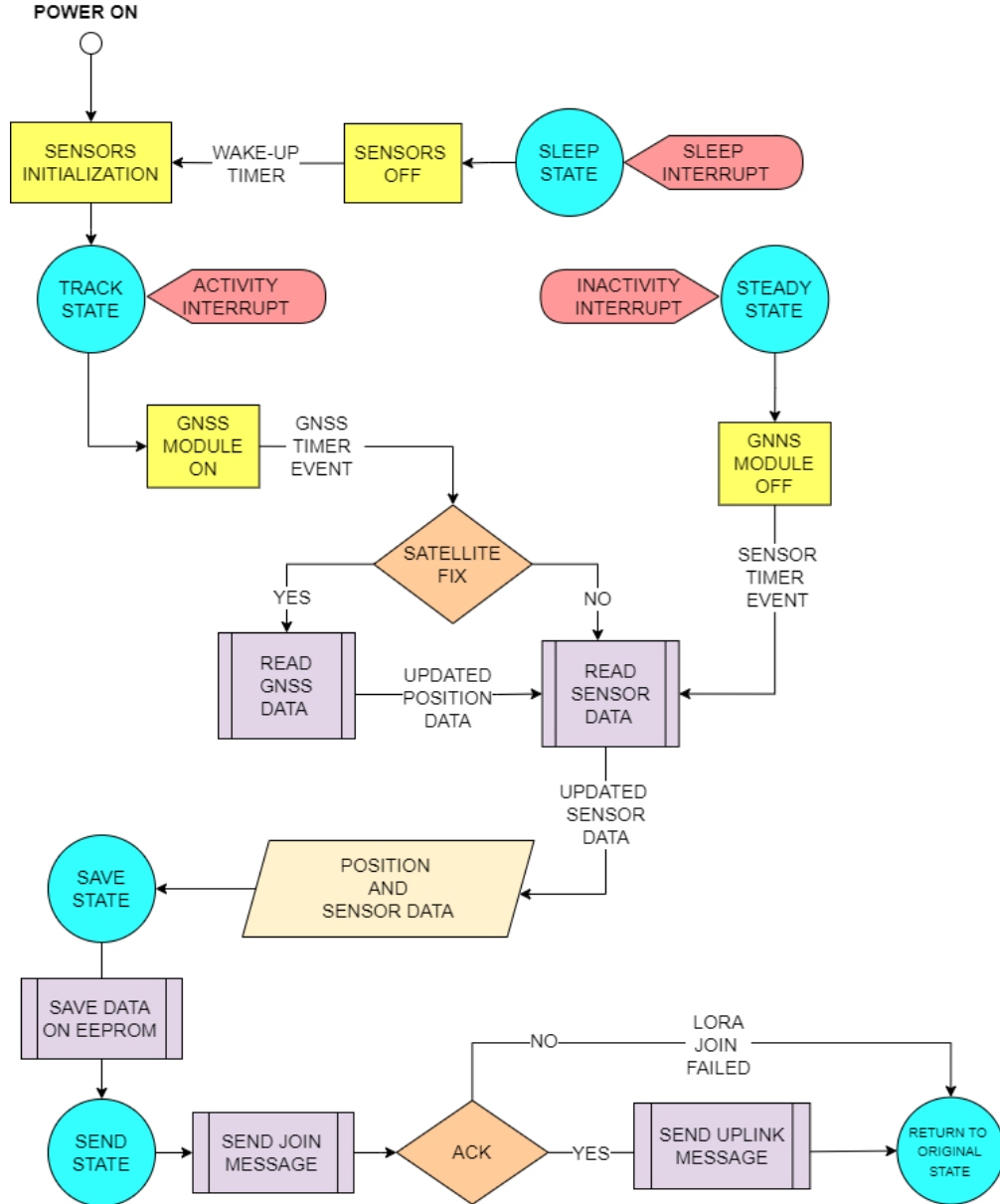


Figure 4.3: Flowchart

- **POWER ON:** turn on the device by keeping the button pressed.
- **SENSOR INITIALIZATION:** temperature, pressure and humidity sensors are powered up as well as the GNSS Teseo Module and the accelerometer.
- **TRACK STATE:** the accelerometer, GNSS module and environmental sensors are powered. The device read and update position and sensors data and send them periodically, depending on TIMER calibration.
- **STEADY STATE:** accelerometer, and environmental sensors are powered. The device read and update only sensors data and send them periodically, depending on TIMER calibration according to different use cases.
- **SLEEP STATE:** All sensors are off. The device wakes up when a calibrated TIMER has expired.
- **SAVE STATE:** the device can store data in the EEPROM memory to track the unit load also when it is not in the range of LoRaWAN network coverage.
- **SEND STATE:** the device send a join message to the LoRaWAN Network, waiting for the ACK message. The device retrieve the last unsent data from the EEPROM and send an uplink message to the application.

4.3.2 Interrupts

The FP-ATR-LORA1 function pack provides a board support package (BSP), the layer of software containing hardware-specific firmware to manage the interrupt pin (Interrupt Service Routine) of the three-axis accelerometer. The accelerometer recognize a movement (Activity Interrupt) and the State Machine switches to Track State. In the default implementation , SysTick timer is the source of time base. It is used to generate interrupts at regular time intervals. The State Machine switches to STEADY STATE because of the INACTIVITY INTERRUPT, when the accelerometer do not send any signal on the interrupt pin for a certain period. The SLEEP INTERRUPT makes the State Machine switch to SLEEP STATE, after a timer event.

4.3.3 GNSS satellite fix

Time to First Fix is the time required for a GNSS device to acquire satellite signal. The device must have three sets of data to get precise positioning data: GNSS satellite signals, almanac data, and ephemeris data. The FP-ATR-LORA1 software package allows to set a max number of attempt. When the max number of attempts is exceeded, the device reads and sends only sensor data.

4.4 Serial Port Communication

The serial communication allows the user to monitor the application status and the state machine evolution and to write control and configuration commands. The serial interface has been used to:

- Set sensor and GNSS read timer events
- Set the LoRa keys and store them on the EEPROM
- Read EEPROM log data
- Debugging

The terminal emulator used for the serial port connection is Tera Term, an open source software maintained by a japanese developer team (Tera Term Project). It was very practical to use also serial communication on an android device to monitor the device during on-field tests.

The screenshot shows the Tera Term Serial Port Configuration window. It contains the following fields and buttons:

- Port:** COM8 (dropdown)
- Speed:** 115200 (dropdown)
- Data:** 8 bit (dropdown)
- Parity:** none (dropdown)
- Stop bits:** 1 bit (dropdown)
- Flow control:** none (dropdown)
- Buttons:** New setting, Cancel, Help
- Transmit delay:** 10 msec/char, 10 msec/line

Figure 4.4: Tera Term Serial Port Configuration

4.4.1 Over The Air Activation

Over The Air Activation procedure is the fastest method to register the device to a LoraWAN service provider. The OTAA join procedure requires the end device to be within the coverage of the network. The FP-ATR-LORA1 function pack application layer allows the user to set the LoRa keys via Serial Port. The parameters to set for join the network are:

- Device Extended Unique Identifier (EUI): provided by device manufacturer.



Figure 4.5: Serial USB terminal Android App

- Join EUI: is a 64 extended unique identifier of the Join Server during activation.
- App Key: encryption key to secure messages which carry a payload
- Network Key: encryption key to secure messages which do not carry a payload

```

VT COM8 - Tera Term VT
Processing command: ?devicejoinparameter

DevEui(FromMcuId) = 37 37 33 31 6D 37 73 07
Application uses the Over-the-Air activation procedure.
Active keys:
DevEui = 37 37 33 31 6D 37 73 07
JoinEui = 70 B3 D5 7E D0 01 73 07
AppKey = F4 7C 1C C2 4E 98 37 08 8D 01 3C D3 C0 71 1B 36
NwkKey = F4 7C 1C C2 4E 98 37 08 8D 01 3C D3 C0 71 1B 36
    
```

Figure 4.6: OTAA LoRa keys

Chapter 5

The Things Network

The Things Network is a global collaborative ecosystem used by thousands of developers and companies around the world that guarantees access to LoRaWAN® services in more than 150 countries. In this thesis project The Things Network has been chosen as decentralize network server to broadcast device messages during on field testing. The Things Network has a strong Italian community and with more than 300 public and private gateways it provides coverage in more than 30 cities [13].

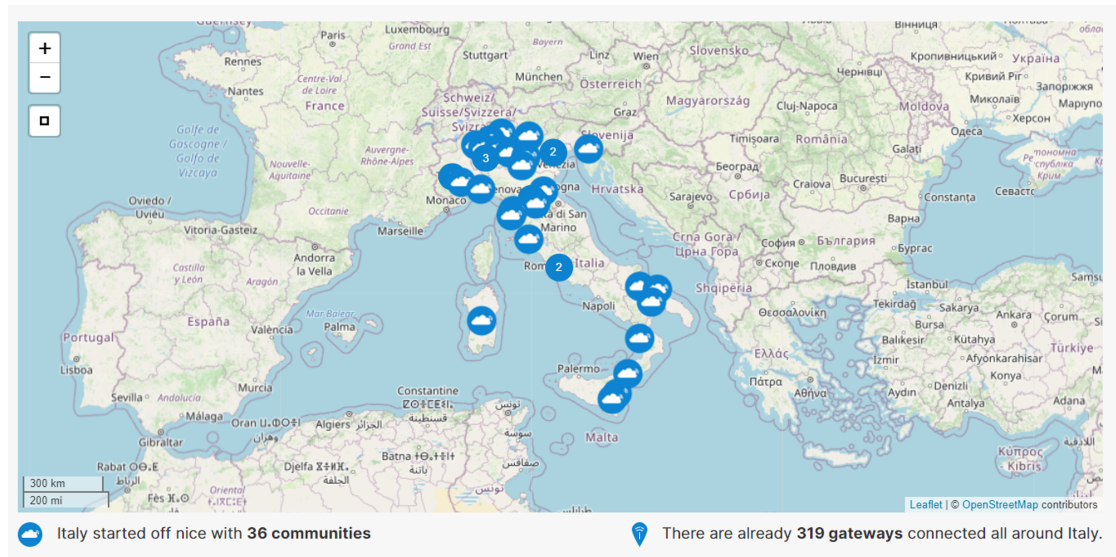


Figure 5.1: The Things Network in Italy

Community	Gateways	Contributors
Bari	1	8
Bergamo	2	8
Bologna	9	25
Brescia	1	18
Calabria	13	9
Catania	1	7
Como	2	9
Cuneo	40	10
Emilia Romagna	21	11
Firenze	1	3
Genova	2	11
Lazio	11	30
Livorno	2	10
Matera	8	4
Messina	2	3
Milano	14	46
Monza	16	12
Mugello	14	5
Roma	4	27
Sardegna	12	14
Trento	6	5
Treviso	2	14
Trieste	6	11
Torino	7	19
Veneto	124	23

Table 5.1: The Things Network in Italy

5.1 Application

The Things Network provides a console to create an Application to manage the device connection to the server, register a private gateway and monitor live data stream.



Figure 5.2: The Things Network Application

5.1.1 Device Registration

The device need to be registered in the application to communicate with the network. In the previous chapter it has been described how to set the keys on the device EEPROM memory. On the Application side the following parameters must be set.

- End Device ID: thesis-tracker
- Frequency plan: Europe 863-870MHz
- LoRaWAN version: LoRaWAN Specification 1.0.3
- DevEUI: 58A0CBFFFE803833
- AppEUI: auto-generated
- AppKey: auto-generated

The authentication keys to connect the device and the application can be manually or automatically generated by the Application.

5.1.2 Gateway registration

The gateways are the bridge between End-Node devices and The Things Network. The devices use LoRaWAN protocol to connect to a gateway, while the gateway is connected to the TTN via Wifi, Ethernet or Cellular protocols. The Things Indoor Gateway (TTIG) has been used to guarantee network coverage to the device during implementation steps. It is designed to be compliant with LoRaWAN protocol,



Figure 5.3: An example of outdoor LoRa Gateway models

powered with USB-C type on 900mA or a wall plug, easy to set-up and can be connected to any network with an open source protocol created by Semtech.



Figure 5.4: The Things Indoor Gateway

5.2 Live Stream Message

The registered device send a LoRaWAN Uplink Messages to the Network Server. The Application console provides the developer the tools for format the payload and have access to live stream messages.

Gateway ID	my-thesis-gateway
Gateway EUI	58A0CBFFFE803833
Frequency plan	Europe 863-870MHz

Table 5.2: TTIG Registration Data

Chipset	Semtech SX1308
Channels	8
Sensitivity	-140/-135 dBm (EU/US)

Table 5.3: TTIG Datasheet

Time	Type		Verbose stream <input type="checkbox"/>	Export as JSON	Pause	Clear
↑ 17:57:36	Forward location solved me...	Latitude: 45.0205 Longitude: 7.7065 Altitude: 596 Source: GPS				
↑ 17:57:36	Update end device	["locations"]				
↑ 17:57:36	Forward uplink data message	DevAddr: 27 FE 0D BF <> Payload: { accelerometer_3: {...}, analog_in_5: 4.				
↑ 17:57:36	Successfully processed dat...	DevAddr: 27 FE 0D BF <>				
↑ 17:55:34	Forward location solved me...	Latitude: 45.0205 Longitude: 7.7065 Altitude: 596 Source: GPS				
↑ 17:55:34	Update end device	["locations"]				
↑ 17:55:34	Forward uplink data message	DevAddr: 27 FE 0D BF <> Payload: { accelerometer_3: {...}, analog_in_5: 4.				
↑ 17:55:34	Successfully processed dat...	DevAddr: 27 FE 0D BF <>				

Figure 5.5: Live Data TTN Application Console

5.2.1 Cayenne Payload

1 Byte	1 Byte	N Bytes	1 Byte	1 Byte	M Bytes	...
Data1 Ch.	Data1 Type	Data1	Data2 Ch.	Data2 Type	Data 2	...

Figure 5.6: Cayenne Payload Format

The device has been configured to send data to The Things Network in line with the Cayenne Low Power Payload format. The payload size can be lowered down to 11 bytes, and allows the device to send multiple sensor data. The payload structure must add two prefixed byte to the data frame:

- Data Channel: 1 byte to identify the sensor
- Data Type: 1 byte to identify the type of data
- Data Frame: N byte depending on type of data

The device send Battery Level, Temperature, Humidity, Position, Accelerometer and Barometer data type. Each data type is characterized by a Object Identifier, Data Size and a Data Resolution per bit.

Data Type	ID (Hex)	Data size (byte)	Data Resolution
Battery Level	2	2	0.01 signed
Temperature	67	2	0.1°C signed MSB
Humidity	68	1	0.5% unsigned
Accelerometer	71	6	0.001g signed MSB per axis
Barometer	73	2	0.1hPa unsigned MSB
GPS location	88	9	Lat:0.0001° signed MSB Lon:0.0001° signed MSB Alt:0.01 m signed MSB

Table 5.4: Cayenne Payload Format

5.2.2 Payload Formatter

The things network has a powerful integrated tool to write Javascript payload formatter for on line conversion of the the binary payload into a human readable JSON object. The decoded payload is a JSON uplink message reporting data of:

00732597016701010268460371ffe603d30003048806de9e012d0600eacd050201a1060100070000

Figure 5.7: Payload Hex format

- Temperature: °C
- Humidity: %
- Pressure: hPa
- Battery: V
- GNNS position: lat, long, alt

```
"decoded_payload": {  
  "accelerometer": {  
    "x": -0.062,  
    "y": 1.004,  
    "z": 0.054  
  },  
  "battery": 4.04,  
  "pressure": 957.3,  
  "gnns": {  
    "altitude": 596.45,  
    "latitude": 45.0205,  
    "longitude": 7.7065  
  },  
  "relative_humidity": 38,  
  "temperature": 33  
}
```

Figure 5.8: Payload JSON format

5.3 Transmission Limitation

Over-head [byte]	Payload [byte]	Data rate	SF	Duty Cycle [msg/h]	Airtime [ms]
25	34	DR4	8	206	174.6

Table 5.5: Airtime Calculated

5.3.1 Duty Cycle

The duty cycle of a device transmission is fraction of time in which the transmission is active over the unit of time. LoRaWAN MAC protocol is limited by regulatory authorities to a 1% duty cycle [14].

5.3.2 TTN Fair Use Policy

This project relay on The Things Network Community that guarantees free access to the shared gateways but with a shorter time, limiting the uplink airtime to 30s/24h per node and the downlink message to 10 msg/24h.

5.4 MQTT Server

The Things Network maintains a MQTT Server (Standard Version 3.1.1, QoS=0) to work with the message stream. An API key has been generated in order to connect to the MQTT Server. TTN provides integrations with the most popular MQTT Client as Eclipse Mosquitto or Eclipse Paho. In this thesis project the MQTT sever has been connected with Node-Red as described in the next chapter.

Chapter 6

IoT platform

6.1 Node-RED

Node-RED is a programming JavaScript based tool to develop IoT solutions. It is built on Node.js and taking full advantage of its event-driven non-blocking model can run on very low-cost devices. For this thesis project Node-RED has been installed on a Raspberry Pi 4 through command line and set to run as a service when the Raspberry is turned on. The Raspberry runs Node-RED h24 and has been upgraded with a fan and heat sinks to improve performance.

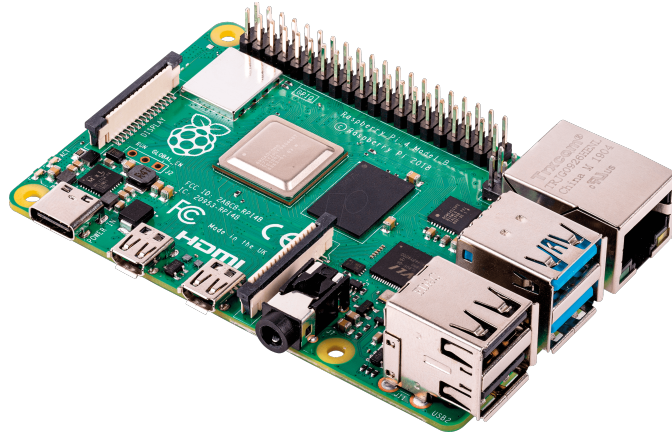


Figure 6.1: Raspberry Pi 4 - 1GB RAM

6.1.1 MQTT Client

The MQTT In Node connects Node-RED to a MQTT broker and subscribes to messages from the specified topic. The output of the MQTT-in node is the uplink

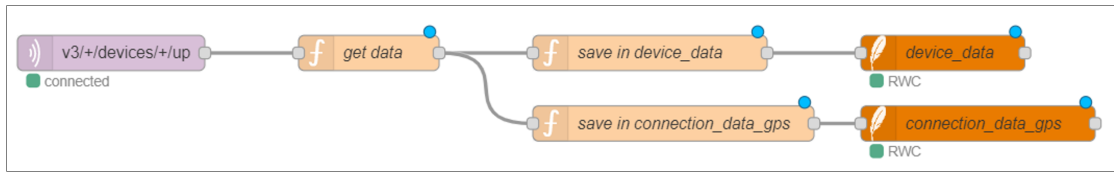


Figure 6.2: MQTT-in node

message as a JSON object.

6.1.2 Message parsing

The uplink message is composed by a:

- end device id: identifier of the device.
- application id: identifier of The Things Network application
- receiving time: arrival time of the device message to The Things Network Server
- decoded payload: device sensor and positioning data
- rx metadata: gateway id and quality of connection data
- network ids: cluster address

The get-data function node saves the device sensor data and position in JavaScript variables when an uplink message arrives. The JSON object is converted in an array of variables that will be stored in the database.

6.1.3 Data Store

The Raspberry Pi 4 can host different databases. Considering the hardware limit, a SQLite database has been installed. Two function has been implemented to save device data and connection data.

6.1.4 Load Status Check

The Load Status Check function has been define to set thresholds on the device sensor data, depending on the application. This function set a Load Alarm boolean variable to true when thresholds are exceeded.

- Temperature Threshold: max and min °C degree values are defined to set the Load Alarm when the Unit Load temperature exceed a defined range.

```

"uplink_message":
  {"session_key_id": "AYHta7QqVRgU9/VF86pP1g==",
   "f_port": 99,
   "f_cnt": 323,
   "frm_payload": "AHM1YwFnATkCaGMDCQAHA1H+PASIBt6aAS0HAOkIBQIBowYBAA==",
  }
"decoded_payload":
  {"accelerometer_3": {"x": 0.007, "y": 0.849, "z": -0.452},
   "analog_in_5": 4.19, "barometric_pressure_0": 957.1,
   "digital_out_6": 0,
   "gps_4": {"altitude": 596.56, "latitude": 45.0202, "longitude": 7.7063},
   "relative_humidity_2": 49.5,
   "temperature_1": 31.3},
  }
"rx_metadata":
  {"gateway_ids": {"gateway_id": "my-thesis-gateway"},
   "time": "2022-07-12T00:09:12.508119106Z",
   "rssi": -113,
   "channel_rssi": -113,
   "snr": -4,
   "location": {"latitude": 44.76538779047523, "longitude": 7.248659133911134},
   "consumed_airtime": 0.174592
  }

```

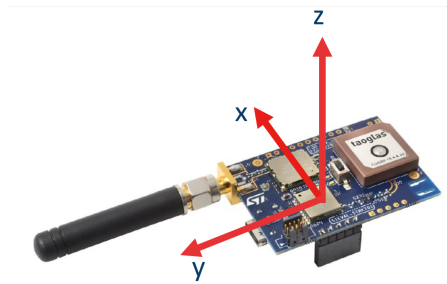
Figure 6.3: Uplink Message

Server	eu2.cloud.thethings.industries:1883:18	
Action	Subscribe to single topic	
Topic	v3/+devices/+up	
QoS	0	
Output	a parsed JSON object	
Name	Name	

Figure 6.4: MQTT-in node configuration

- Humidity Threshold: max and min humidity % values are defined to set the Load Alarm when the Unit Load humidity exceed a defined range.
- Accelerometer Threshold: max and min g values on x,y,z axis to detect if the device has sensed shocks. The pitch and roll angles are calculated to

Variables	Description
var time	Timestamp of the message
var gat_id	Gateway ID
var gssi	Received Signal Strength Indicator
var snr	Signal to Noise Ratio
var airtime	Consumed air time
var band	LoRa bandwidth
var sf	LoRa spreading factor
var c_rate	LoRa coding rate
var gat_lat	Gateway position, latitude
var gat_lon	Gateway position, longitude
var temp	Device temperature sensor data
var hum	Device relative humidity sensor data
var acc_x	Device accelerometer x axis data
var acc_y	Device accelerometer y axis data
var acc_z	Device accelerometer z axis data
var lat	Device GNSS position latitude data
var long	Device GNSS position longitude data

Table 6.1: Variables from Uplink message**Figure 6.5:** Device axis

determine the orientation of the Unit Load in the space and set an alarm

when a maximum tilt angle is exceeded.

$$PitchAngle = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \quad (6.1)$$

$$RollAngle = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \quad (6.2)$$

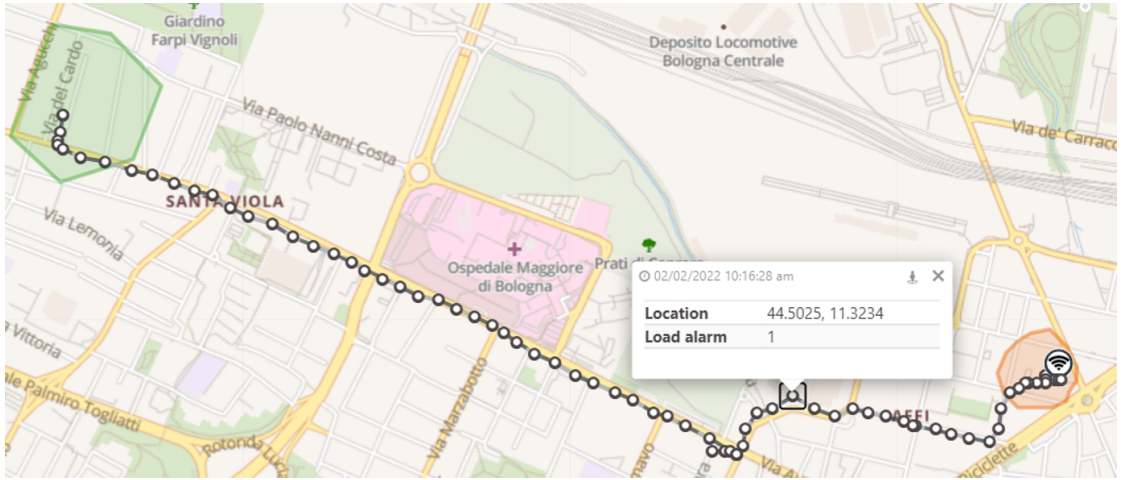


Figure 6.6: Geofence

6.1.5 Geofencing

A Geofence is a virtual perimeter defined by geographic coordinates that surround a physical area. When a device locates itself inside or outside the Geofence boundary can trigger different actions. Geofencing is useful on tracking a Unit Load along the supply chain to identify warehouses and know in real time if the device is inside or outside. In this project the geofences are defined by JavaScript function that send a notification (or email) to the user when the device cross the perimeter.

6.1.6 Dashboard

A Dashboard has been implemented to monitor the device sensor data, have real-time access to the device position and retrieve data from the database. The Dashboard is composed by:

- Humidity, Temperature, Tilt and Battery Level indicators

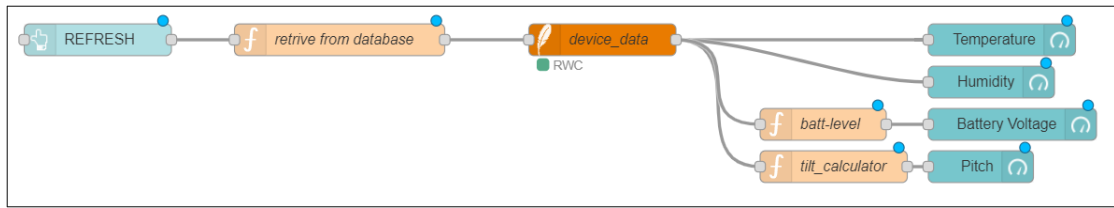


Figure 6.7: Dashboard Node-RED flow

- A map to see real-time positioning of the device
- A graphs section to visualize the sensors samples over time

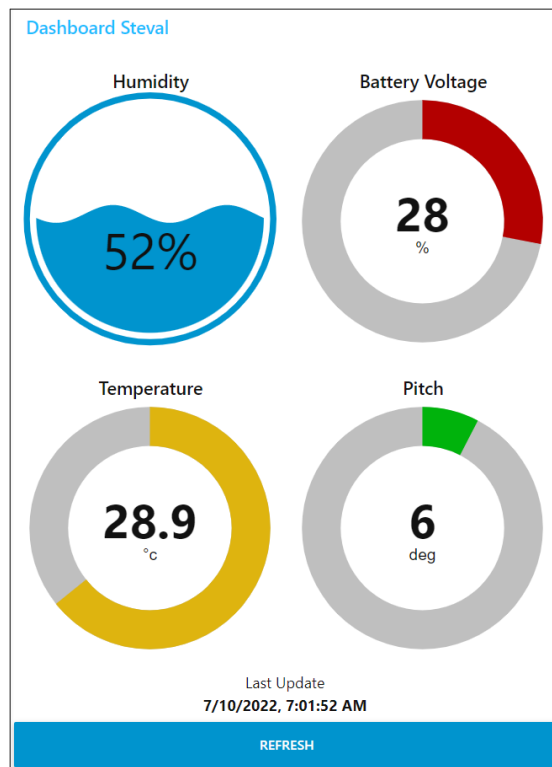


Figure 6.8: Dashboard Indicators

6.1.7 Reporting and Notification

An automatic e-mail sending feature has been implemented to report the distance travelled by the Unit Load together with sensors data and load alarm geographic coordinates. Node-RED allows the triggering of function node through Telegram

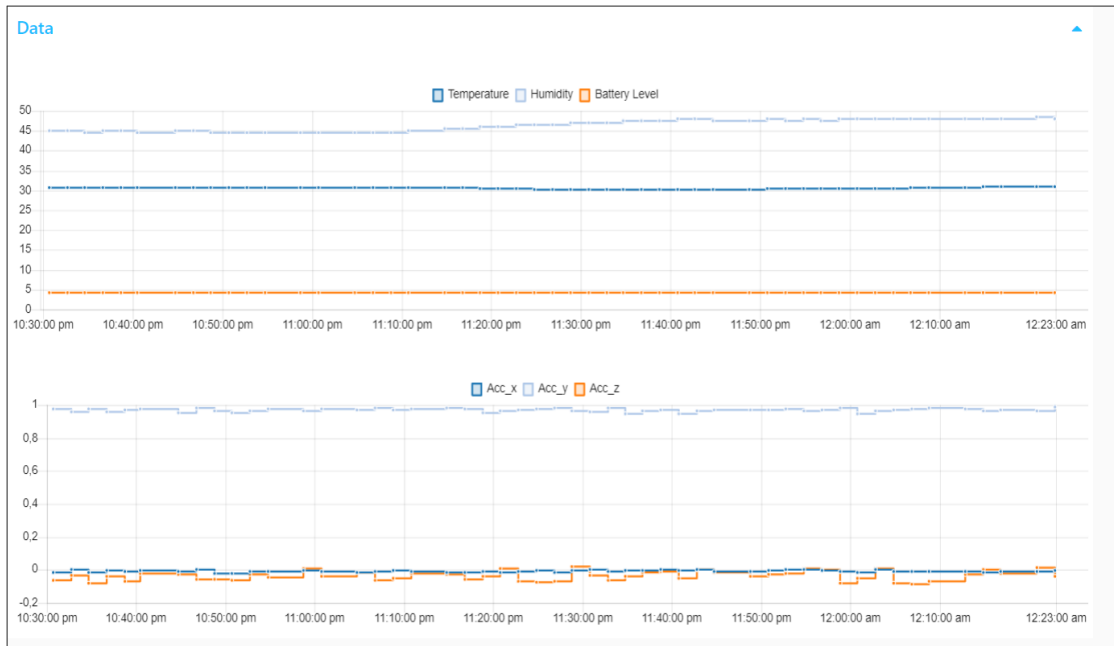


Figure 6.9: Dashboard graphs

bot command. A user can access to device position and sensors data on his personal device, adding the bot on the Telegram app.

6.2 Database

The SQLite database is made by two tables:

- Device Data: Sensor data, position data and the time of arrival of the message are stored in a table.
- Connection Data: Gateway ID, RSSI, SNR, Air Time, Bandwidth, SF, Code rate, Gateway latitude, Gateway longitude and the time of arrival.

Device Data												
ID	TEMP	HUM	BATT	ACC_X	ACC_Y	ACC_Z	LOAD_A...	LAT	LONG	ALT	TIMEST...	
6938	32.9	43.5	4.13	0.005	0.961	-0.032	0	45.0204	7.706	596.37	16575626326...	
6937	32.9	43.5	4.12	0.004	0.954	-0.044	0	45.0204	7.706	596.37	16575622680...	
6936	32.9	43	4.11	0.007	0.964	-0.023	0	45.0204	7.706	596.37	16575621466...	
6935	32.9	43.5	4.1	0.005	0.953	-0.05	0	45.0204	7.706	596.37	16575620250...	
6934	32.9	43.5	4.09	0.002	0.957	-0.033	0	45.0204	7.706	596.37	16575617821...	
6933	32.9	43.5	4.08	0.005	0.956	-0.038	0	45.0204	7.706	596.37	16575617130...	
6932	32.9	43.5	4.09	0.009	0.961	-0.031	0	45.0204	7.706	596.37	16575615919...	
6931	33	43	4.1	0.001	0.97	-0.039	0	45.0204	7.706	596.37	16575613485...	
6930	32.9	43.5	4.08	0.002	0.957	-0.034	0	45.0204	7.706	596.37	16575612273...	
6929	33	43.5	4.07	0.002	0.959	-0.027	0	45.0204	7.706	596.37	16575608937...	
6928	33.1	43.5	4.07	-0.005	0.965	-0.04	0	45.0204	7.706	596.37	16575607723...	
6927	33.1	43	4.06	0.003	0.962	-0.039	0	45.0204	7.706	596.37	16575606507...	
6926	33.1	43	4.03	0.005	0.961	-0.034	0	45.0204	7.706	596.37	16575604073...	
6925	33.1	43	4.01	0.007	0.962	-0.039	0	45.0204	7.706	596.37	16575602858...	
6924	33.2	43	4	0	0.952	-0.031	0	45.0204	7.706	596.37	16575600429...	
6923	33.2	43.5	4.02	0.008	0.961	-0.024	0	45.0204	7.706	596.37	16575599213...	
6922	33.1	43	3.98	0.007	0.949	-0.035	0	45.0204	7.706	596.37	16575597998...	
6921	33.3	43	3.97	0.009	0.954	-0.045	0	45.0205	7.706	596.4	16575594364...	
6920	33.2	43.5	3.98	0.005	0.958	-0.041	0	45.0205	7.706	548.21	16575593151...	
6919	33.2	43.5	3.99	0.002	0.964	-0.032	0	45.0204	7.706	596.37	16575591933...	
6918	33.3	43.5	3.97	0.013	0.942	-0.024	0	45.0204	7.706	596.37	16575590719...	
DOWNLOAD												

Figure 6.10: Device Data retrieved from the Database

Gateway Data												
ID	GATEWA...	RSSI	SNR	AIR_TIME	BANDWI...	SF	CODE_RATE	GAT_LAT	GAT_LON	TIMESTAMP		
1040	packetbroker	-113	-7.5	0.174592s	125000	8	0	44.765387790...	7.2486591339...	1654936628597		
1039	my-thesis-gate...	-93	11.5	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654936508537		
1038	my-thesis-gate...	-90	8.5	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654936388433		
1037	my-thesis-gate...	-93	9.25	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654936268347		
1036	my-thesis-gate...	-94	11	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654936148255		
1035	packetbroker	-115	-11.5	0.174592s	125000	8	0	45.014941981...	7.7021884918...	1654936028166		
1034	my-thesis-gate...	-93	8.5	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654935908114		
1033	my-thesis-gate...	-90	11.25	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654935787989		
1032	my-thesis-gate...	-93	11.5	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654935667912		
1031	my-thesis-gate...	-91	11.25	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654935547830		
1030	packetbroker	-115	-9.25	0.174592s	125000	8	0	45.014941981...	7.7021884918...	1654935427949		
1029	my-thesis-gate...	-95	7.75	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654935307694		
1028	my-thesis-gate...	-95	10.5	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654935187756		
1027	my-thesis-gate...	-93	11	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654935067645		
1026	my-thesis-gate...	-96	10	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654934947555		
1025	my-thesis-gate...	-91	10.75	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654934827450		
1024	my-thesis-gate...	-93	11.5	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654934707367		
1023	my-thesis-gate...	-94	1	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654934587247		
1022	packetbroker	-113	-7.25	0.174592s	125000	8	0	45.014941981...	7.7021884918...	1654934467070		
1021	my-thesis-gate...	-93	8.25	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654934346932		
1020	my-thesis-gate...	-100	10.5	0.174592s	125000	8	0	45.020572850...	7.7061125636...	1654934226839		
DOWNLOAD												

Figure 6.11: Connection Data retrieved from the Database

Chapter 7

Test and Experiments

Several outdoor tests have been performed to verify the functioning of the device. Two different scenarios have been analyzed and reported.

7.1 TEST A: Mid-haul shipping through a non-urban environment

The device has been placed into a shipping box to simulate a real working condition and delivered from Alba (Piedmont) to Turin along the SR29 regional road. In this particular scenario GNSS read and sensor polling timer has been set to 15min: the device will send a message every 15 min, a good trade-off to have valuable position info and a smooth battery discharge curve.

Temperature	Humidity	Tilt
30	80%	5°

Table 7.1: Load Alarm Sensor Thresholds

Distance [km]	Type of transport	Duration time [HH:mm:ss]	Road Traffic
49,7	vehicle	02:00:59	High

Table 7.2: Test Description

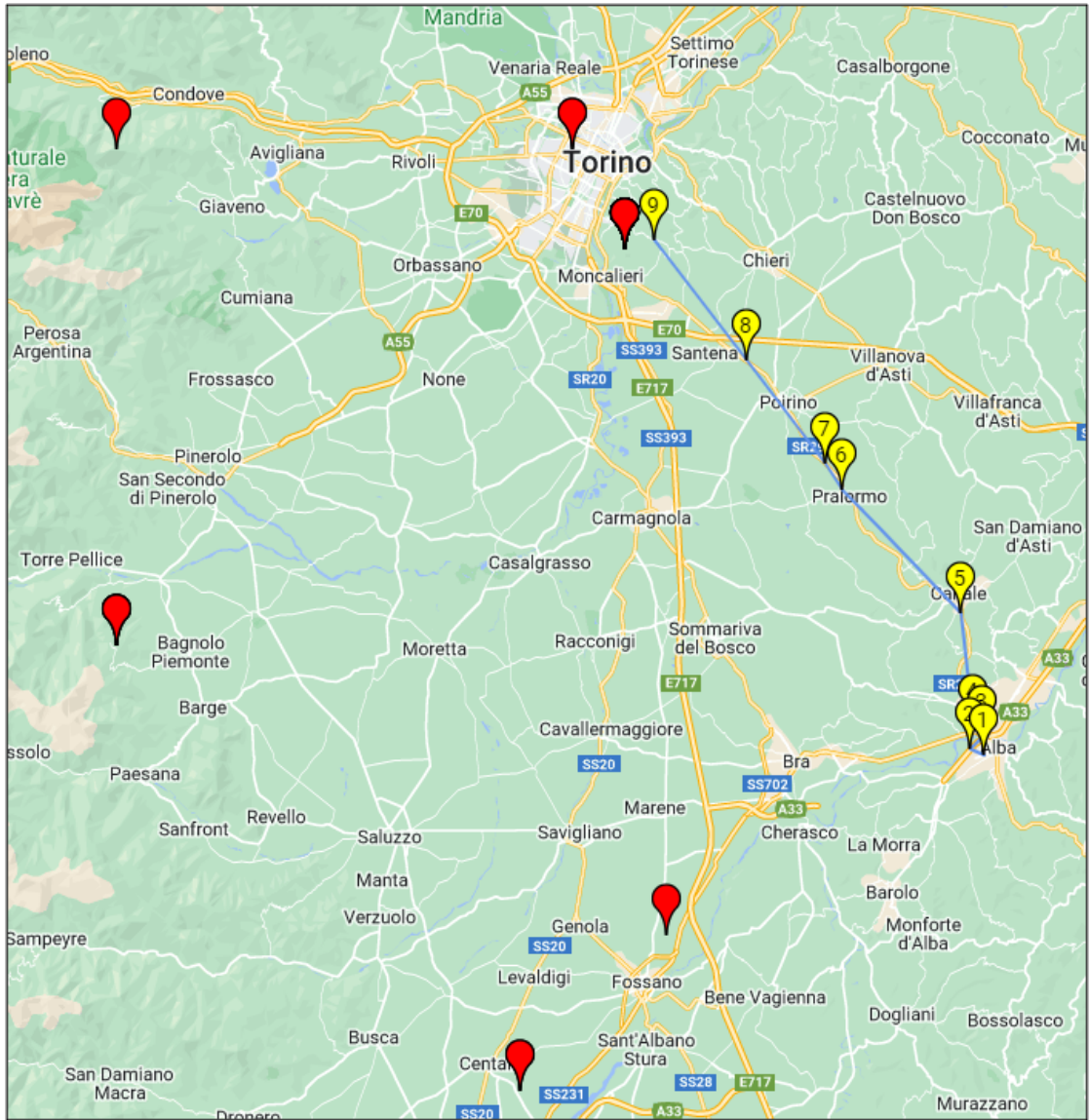


Figure 7.1: Device position (yellow) and Gateways position (red)

Message number	Payload [byte]	Load Alarm	RSSI	SNR	Airtime Consumed [s]	Received Time
1	34	Tilt	-115	-5.25	0.174592	08-May-2022 07:08:02.577
2	34	0	-112	-2.75	0.174592	08-May-2022 07:24:04.898
3	34	0	-111	-8.75	0.174592	08-May-2022 07:39:09.132
4	34	0	-68	11.5	0.174592	08-May-2022 07:54:01.132
5	34	0	-109	0.25	0.174592	08-May-2022 08:09:04.278
6	34	0	-101	-1.25	0.174592	08-May-2022 08:24:01.462
7	34	0	-114	-5.5	0.174592	08-May-2022 08:39:01.462
8	34	Temp	-112	-3.75	0.174592	08-May-2022 08:54:01.462
9	34	0	-120	-5.5	0.174592	08-May-2022 09:09:01.462
TOT	306	2	-107 (avg)	-2.33 (avg)	1.57	00:15:01 time inter- val

Table 7.3: Message data

Gateway EUI	Lat	Lon	Msg	Dist[km]
60C5A8FFFE78FD65	44.5814	7.7390	1	25.2
60C5A8FFFE78FD65	44.5814	7.7390	2	24.8
60C5A8FFFE78FD65	44.5814	7.7390	3	25.3
60C5A8FFFE78FD65	44.5814	7.7390	3	25.4
60C5A8FFFE7664C6	44.4182	7.6094	5	51.9
60C5A8FFFE78FD47	44.7653	7.2486	6	52.4
B827EBFFFE6F9FEE	45.0709	7.2467	7	54.95
60C5A8FFFE7664C3	45.0149	7.7021	8	11.63
60C5A8FFFE7664C3	45.0149	7.7021	8	0.76

Table 7.4: Gateways

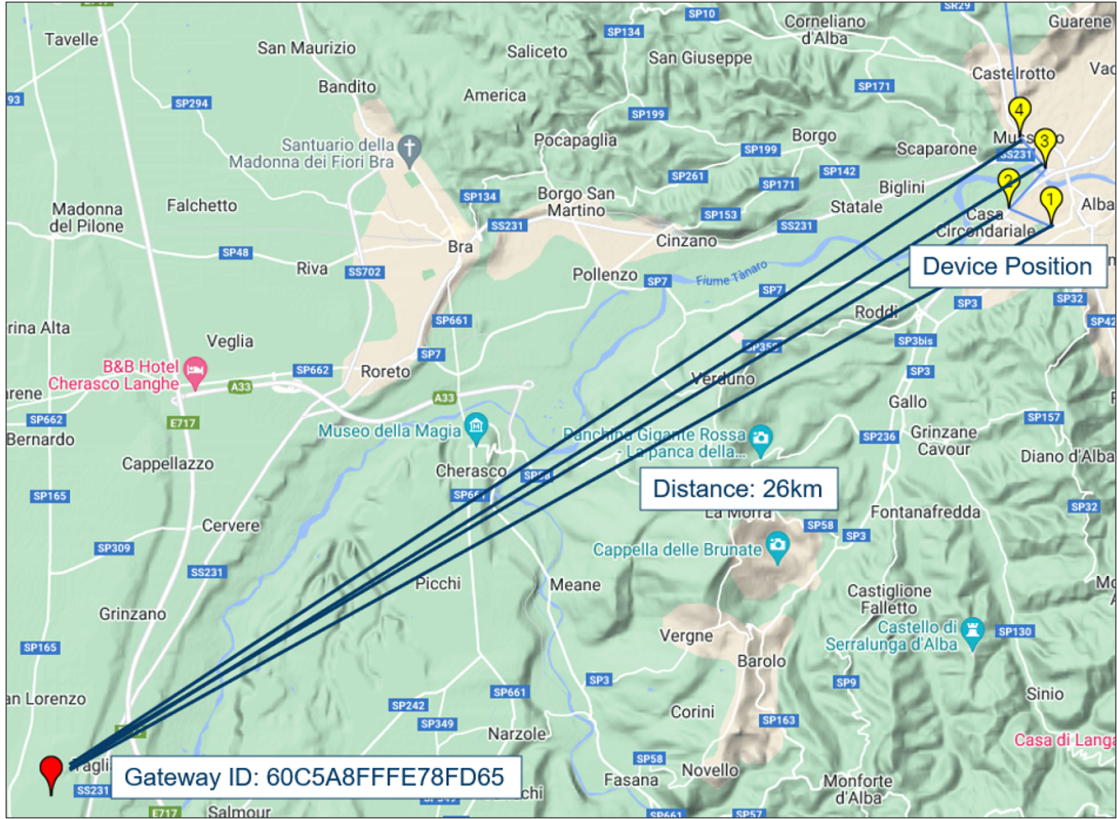


Figure 7.2: Distance Device - Gateway

The received data show that 2 load alarms have been set during the test: A Tilt alarm corresponding to the loading procedure on the vehicle and an exceeding of the Temperature threshold.

Load Alarm	Latitude	Longitude
Temperature	44.9449	7.811
Tilt	44.6963	8.0227

Table 7.5: Threshold position

7.2 TEST B: Urban delivery on Cargo Bike

The device has been placed in a shipping box and loaded on a Cargo bike simulating a delivery in a urban area. In this particular scenario the GNSS read and sensor polling timer has been set to 20s to track more frequently the Unit Load considering the shorter distance and the more stressed condition.



Figure 7.3: cargo bike set-up

Temperature	Humidity	Tilt
30	80%	5°

Table 7.6: Load Alarm Thresholds

Distance [km]	Type of transport	Duration time [HH:mm:ss]	Weather
6,4	cargo-bike	00:09:32	Cloudy

Table 7.7: Test Description

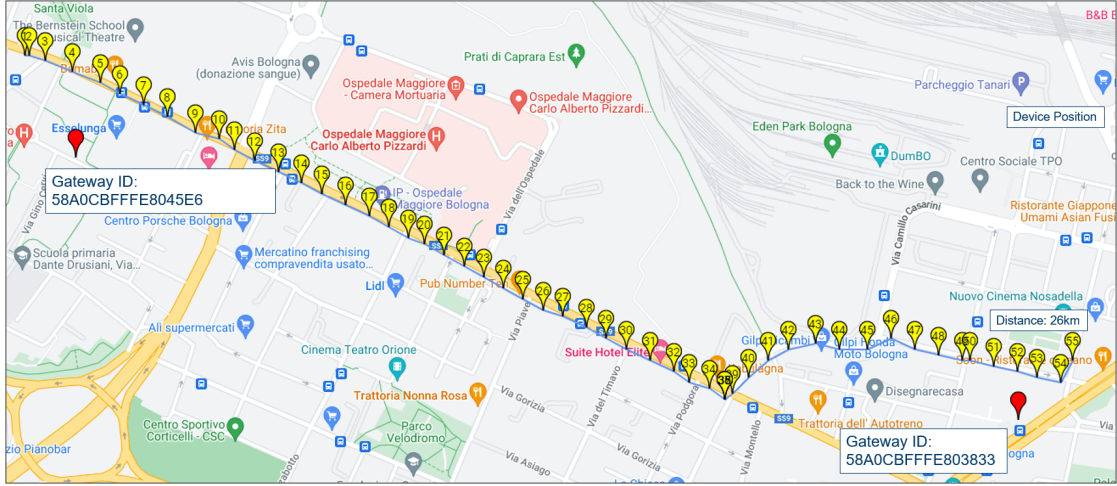


Figure 7.4: Device position (yellow) and Gateways position (red)

Msg tot	Payload tot [byte]	Load Alarm	RSSI [avg]	SNR [avg]	Airtime Consumed [s]	Time Interval [avg]
55	1.870	Hum(23) Tilt(5)	-112	-6.8	9.59	00:00:20.898

Table 7.8: Message data

Load Alarm	Start	Stop
Humidity	44.501472, 11.320668	44.500385, 11.324005
Tilt	44.502076, 11.319241	44.501173, 11.330420

Table 7.9: Threshold Position

Chapter 8

Conclusion and Future Work

8.1 Conclusion

It has been investigated that the use of a low-price device to monitor the delivery of a unit load can provide important information on the quality of transport service, detecting the geographic location of the stresses suffered by a Unit Load, respecting the duty cycle limits imposed by the legislation. For this research project, access to the LoRaWAN network via The Things Network in the Community version lead to good results. For an industrial-scale deployment of tracking devices it may not be sufficiently performing, but the LoRaWAN network in Italy is growing making this kind of IoT solution more and more widespread.

8.2 Future Work

This system can evolve to a more complete version by implementing an indoor tracking system based on the RSSI and the Time of Arrival (ToA) of the signal. The gateways installation inside a storage areas can be exploit to triangulate the signal and define the position of the device with a certain precision. This implementation would make the tracking system effective even indoors, where the GNSS signal can be strongly disturbed.

Bibliography

- [1] Statista. *Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030*. (2022). Accessed: April. 11 2022. [Online]. URL: <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide> (cit. on pp. 1, 2).
- [2] Federica Laricchia. *Market size of Industry 4.0 in Italy from 2017 to 2019 with a forecast for 2020 to 2024*. (2022). Accessed: March. 15 2022. [Online]. URL: <https://www.statista.com/statistics/1057237/industry-four-point-zero-market-size-italy> (cit. on pp. 3, 4).
- [3] ASQ. *ASQ QUALITY 4.0 SUMMIT*. (2021). URL: <https://asq.org/conferences/quality-4-0/2021> (cit. on p. 2).
- [4] Prescient Strategic Intelligent. *IoT in Logistics Market*. Report. Prescient Strategic Intelligent, 2020 (cit. on p. 4).
- [5] AllTheResearch. *Global IoT in Logistics Market - Segment Analysis, Opportunity Assessment, Competitive Intelligence, Industry Outlook 2016-2026*. Report. Prescient Strategic Intelligent, May 2020 (cit. on p. 5).
- [6] Sergio Terzi Oyeol Emmanuel Ebikake Claudio Sassanelli. «PSS design through Design for Supply Chain: State of the art review». In: *ScienceDirect* (2018) (cit. on p. 6).
- [7] Jibum Kim Dong-Hoon Kim Eun-Kyu Lee. «Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed». In: *ResearchGate* (2019) (cit. on p. 10).
- [8] LoRa Alliance. *LoRaWAN® World Expo's In-Depth Technical Sessions Prepare Businesses for Success*. (2022). Accessed: April. 4 2022. [Online]. URL: <https://resources.lora-alliance.org/lora-alliance-blog/lorawan-world-expo-s-in-depth-technical-sessions-prepare-businesses-for-success> (cit. on p. 11).
- [9] Semtech. *What are LoRa and LoRaWAN?* (2022). URL: <https://lora-developers.semtech.com/> (cit. on p. 13).

- [10] The Things Stack. *Reference*. (2022). URL: <https://www.thethingsindustries.com/docs/reference/> (cit. on p. 14).
- [11] STMicroelectronics. *Getting started with the STM32Cube function pack for IoT tracker node with LoRa® connectivity, GNSS and sensors*. UM2541. (2019). URL: https://www.st.com/resource/en/user_manual/um2541 (cit. on p. 20).
- [12] STMicroelectronics. *Getting started with the STEVAL-STRKT01 LoRa® IoT tracker*. UM2487. (2021). URL: <https://www.st.com/en/embedded-software/fp-atr-lora1> (cit. on p. 24).
- [13] The Things Stack. *Quick Start*. (2022). URL: <https://www.thethingsnetwork.org/docs/quick-start/> (cit. on p. 29).
- [14] European Standard. *ETSI EN300.220 standard*. V2.4.1. URL: <https://www.etsi.org/> (cit. on p. 36).